SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Animacija u 2D računalnoj grafici

Valentin Berger

Voditelj: Izv. prof. dr. sc. Domagoj Jakobović

Zagreb, ožujak, 2016.

**Sadržaj**

[1. Uvod 1](#_Toc451367316)

[2. Postupak animacije 2](#_Toc451367317)

[3. Kinematika 3](#_Toc451367318)

[3.1 Hijerarhijska struktura 3](#_Toc451367319)

[3.2 Direktna kinematika 4](#_Toc451367320)

[3.3 Inverzna kinematika 5](#_Toc451367321)

[4. Interpolacija i krivulje 10](#_Toc451367322)

[5. Dodatak 11](#_Toc451367323)

[5.1 Tipovi zglobova 11](#_Toc451367324)

[6. Zaključak 14](#_Toc451367325)

[7. Literatura 15](#_Toc451367326)

[8. Sažetak 16](#_Toc451367327)

# Uvod

Računalna animacija danas je vjerojatno najvažniji aspekt računalne grafike jer ju pronalazimo gotovo svugdje oko nas, od filmova, video igara, crtića pa sve do tv reklama. Jednostavnije, gdje god postoji potreba za prezentacijom nečega, postoji potreba i za računalnom animacijom. No, što je to računalna animacija? Računalna animacija je postupak kojim se statični objekt virtualno pretvara u dinamični objekt, odnosno objekt koji se kreće. Virtualno stoga što je za ostvarivanje animacije potrebno prevariti ljudsko oko, odnosno mozak, tako da se statični objekt prikaže barem 12 puta u sekundi (engl. *frame per second*) koliko je potrebno mozgu da zaključi da se određeni objekt kreće.

Tri su pristupa pri ostvarenju animacije nekog objekta: 1) korištenje fizičkih modela; 2) proceduralna animacija; 3) korištenje ključnih trenutaka/položaja (engl. *keyframing*). Korištenje fizičkih modela je korišten u početcima filmografije i to tako da bi se slikao fizički model nekog objekta u različitim pozicijama te bi se tad spajanjem sličica postigla animacija. Proceduralna animacija se obično koristi u onim situacijama gdje je gibanje nekog tijela ili čestica zasnovano na fizici (npr. simulacija kretanje fluida ili dima vatre). Ovaj seminarski rad će se pozabaviti trećim pristupom, animacijom zasnovanom na definiranju ključnih trenutaka i interpolacije između njih. Takav način animacije je danas je najrasprostranjeniji, a vuče korijene iz razdoblja kada se je animacija ostvarivala ručnim crtanjem bezbroj sličica od kojih je svaka činila jedan ključni trenutak animacije.

# Postupak animacije

Kako je u uvodu rečeno ostvarivanje animacije biti će ostvareno definiranjem ključnih trenutaka (engl. *keyframes*) i interpolacijom između njih. No, postavlja se pitanje nad čime definirati ključne trenutke i što interpolirati? Odgovor na to pitanje možemo pronaći u kinematici koja opisuje kretanje objekata i sustava objekata (kinematički model) ne uzimajući u obzir njihovu masu nego samo geometriju potrebnu da bi se taj objekt iz jednog položaja premjestio u drugi. Zato prvo moramo definirati hijerarhijsku strukturu koju ćemo primjenom kinematike animirati. Za potrebe animacije objekata najbolje je odabrati stablastu strukturu u kojoj čvorovi (engl. *node*) stabla čine segmente (engl. *links*) iliti kosti (engl. *bones*), a veze između čvorova predstavljaju zglobove (engl. *joints*). Takvo stablo je usmjereno i acikličko (engl. *directed acyclic graph*). Nakon definiranja kostura (engl. *skeleton*) potrebno je svakom segmentu dodijeliti mišiće i kožu, tj. u našem slučaju mrežu poligona (engl. *mesh*) na koju ćemo zatim lijepiti teksture, a koji će se pomicati s kosturom. Svaki vrh takvog poligona može biti vezan za nekoliko segmenata s određenom težinom pa svaka transformacija kosti/segmenta je ujedno i transformacija tog vrha. Kako bi se pojednostavnilo definiranje ključnih trenutaka biti će potrebno koristiti inverznu kinematiku, a što će biti detaljnije objašnjeno i sljedećem poglavlju. Nakon što su definirani svi ključni trenutci potrebno je imati mogućnost odrediti položaj kostura u trenutku koji se nalazi između dva ključna trenutka. Za to će nam poslužiti interpolacija definirana preko krivulja, a o čemu će također biti riječ kasnije.

# Kinematika

Kinematika je grana mehanike i matematike kojom se opisuje i gibanje nekog objekta ne uzimajući u obzir utjecaj sila i mase na njegovo gibanje. Gibanje objekta se opisuje kinematičkim jednadžbama čije varijable obično opisuju translacijsko i rotacijsko gibanja (kut, kutna brzina, pozicija, translacijska brzina itd.), a sam objekt mora biti na neki način strukturiran prije početka opisivanja gibanja.

## Hijerarhijska struktura

Zbog potrebe za što prirodnijim načinom ostvarivanja animacija, za svaki objekt koji se animira potrebno je definirati kostur (engl. *skeleton*) kao hijerarhijsku strukturu. Kostur se sastoji od segmenata/kostiju (engl. *bones*) i zglobova (engl. *joints*). Segment je definiran svojom duljinom (udaljenost između dva susjedna zgloba), a kut između dva susjedna segmenta definiran je u zglobu. No, u postupku animacije segment/kost nema niti jednu drugu ulogu osim definiranja udaljenosti između dva krajnja zgloba pa se stoga segment i zglob mogu spojiti tako da se taj pomak između dva susjedna zgloba pohrani u hijerarhijski niži zglob. Sada, nakon navedene transformacije svaki kostur možemo prikazati kao stablo gdje svaki zglob čini jedan čvor stabla (Slika 1).

Slika 1 – Stablasti prikaz kostura

Svaki zglob može imati do 6 stupnjeva slobode (engl. *degree of freedom(DOF)*) u 3-dimenzionalnom prostoru, odnosno do 3 stupnja slobode u 2-dimenzionalnom prostoru, i to na sljedeći način: jedan translacijski stupanj slobode po svakoj od dimenzija (engl. *Prismatic Joint*) te po jedan rotacijski stupanj slobode za svaku ravninu koordinatnog sustava (engl. *Revolute Joint*). Dimenzije zglobova su zanemarive, a složeniji zglobovi se mogu ostvariti povezivanjem više 1DOF zglobova razmaknutih za udaljenost 0. Osim stupnjeva slobode svaki zglob mora imati još definiran i odmak od svojeg zgloba roditelja, ograničenja u zglobovima (minimalne i maksimalne dozvoljene vrijednosti po svakom stupnju slobode). Nakon što smo tako definirali kostur preostaje nam omogućiti neki način specificiranja vrijednosti u zglobovima kako bi odredili pojedine ključne trenutke animacije.

## Direktna kinematika

Nakon što imamo definiran ključni trenutak kostura, tj. određene sve parametre u svim zglobovima, zanima nas koja je pozicija svakog pojedinog zgloba u odnosu na globalni koordinatni sustav u kojemu se kostur nalazi. Odnosno, zanima nas matrica transformacije svakog pojedinog zgloba koja transformira iz lokalnog koordinatnog sustava zgloba u globalni sustava kostura pošto su parametri svakog pojedinog zgloba kostura definirani lokalno, u koordinatnom sustavu zgloba roditelja. Za računanje toga koristi ćemo direktnu kinematiku na sljedeći način:

1. Krenemo obilaziti stablo (kostur) u dubinu (engl. *depth-first search*) počevši od korijenskog čvora.
2. Za svaki čvor (zglob) izračunamo matricu transformacije W:  
   gdje je oblik matrice **L** ovisi o tipu zgloba (vidi 5.3).
3. Obavimo ostale operacije s objektom na temelju izračunatih globalnih matrica transformacija.

## Inverzna kinematika

U složenim kosturima vrlo je teško postaviti kostur u određenu pozu samo manipulirajući ručno zglobovima. Recimo da odaberemo jedan zglob kostura kojeg ćemo nazvati manipulator i da želimo vrh tog manipulatora dovesti u određenu poziciju i orijentaciju u globalnom koordinatnom sustavu. Kako bi smo to uspjeli napraviti moramo znati parametre (kutove, translacije itd.) u svim zglobovima precima tog manipulatora. Te parametre možemo postavljati ručnim podešavanjem svakog zgloba počevši od korijena, što je mukotrpno i neproduktivno, ili pak možemo iskoristiti inverznu kinematiku da to napravi umjesto nas. Znači, ideja inverzne kinematike je izračunati parametre koji moraju biti u zglobovima precima da bi se vrh manipulatora nalazio u nekoj našoj traženoj poziciji i orijentaciji. Inverzna kinematika ima je dobila zato što izračunava obrnuto nego direktna kinematika, a koja nam na temelju parametara u zglobovima precima izračunava globalnu poziciju i orijentaciju vrha manipulatora. Matematički gledano to se može zapisati na sljedeći način:

Postavlja se pitanje kako izračunati funkciju **F**-1(), odnosno kako riješiti sustav nelinearnih jednadžbi? Analitičko rješavanje problema bilo bih najkvalitetnije kada bi uvijek postojalo točno jedno rješenje sustava, što za veće kinematičke lance nije slučaj jer u mnogim slučajevima može postojati puno rješenja, a nekad se može dogoditi da rješenja niti nema. Također, u puno slučajeva analitičko rješavanje nekih nelinearnih sustava je prosto nemoguće, stoga je nužno pristupiti numeričkom načinu rješavanju nelinearnih sustava jer je ono općenito primjenjivo. Rješavanje sustava nelinearnih jednadžbi je optimizacijski problem za koje postoje razne metode rješavanja od kojih većina uključuje izračunavanje Jakobijeve matrice (Jakobiana) i njenog inverza, no postoje neke heurističke metode poput CCD-a (engl. *Cyclic-Coordinate Descent*) koji je specijalno namijenjen za rješavanje problema inverzne kinematike.

Numerička metoda rješavanja navedenog sustava zasniva se na pokušaju linearizacije nelinearnog problema uvođenjem Jakobiana kao linearnog operatora:

Njegovim inverzom mogao bi se izračunati pomak ka konačnom rješenju sustava koji je definiran s pozom **g** traženog manipulatora:

te bi iterativnim ponavljanjem postupka uz dovoljno malen β došli do konačnog rješenja **g**:  
Postavlja se nekoliko pitanja: kako izračunati J na temelju konfiguracije zglobova, kako izračunati inverz J-1 matrice J pošto ona često nije kvadratna, kako odrediti korak integracije β, te kako odrediti kada prekinuti iteriranje? Pa krenimo redom.

Slika 2 – Pseudo kod inverzne kinematike

dok (**e** nije dovoljno blizu **g**) {

Izračunaj **J**(**Φ**) za trenutnu poziciju;

Izračuna **J**-1;

∆**e** = β\*(**g** - **e**);

∆**Φ** = **J**-1 \* ∆**e**;

**Φ** = **Φ** + ∆**Φ**;

Izračunaj novi **e** za **Φ** uz pomoć direktne kinematike;

}

Pošto Jakobijeva matrica opisuje kako promjena u zglobovima utječe na kretnju vrha manipulatora, nužno je za svaki zglob u kinematičkom lancu pojedinačno odrediti kako njegova kretnja (promjena DOF parametara) utječu na kretnju vrha manipulatora (promjenu pozicije i orijentacije). Taj opis čini jedan stupac matrice. Za ilustraciju uzmimo da imamo 1-DOF zglob koji se može rotirati oko osi zadane vektorom **a** i da imamo 3-DOF vrh manipulatora (tražimo samo poziciju u 3-dimenzionalnom prostoru, bez orijentacije). Prilikom računanja utjecaja zgloba na manipulator potrebno je da i zglob i manipulator budu u istom koordinatnom sustavu, najbolje u globalnom koordinatnom sustavu kostura (engl. *world space*). Utjecaj takvog zgloba na manipulator može se računati na način:

, a Jakobijeva matrica bi se popunila na slijedeći način:

Za drugačije tipove zglobova i manipulatora izračunavanje utjecaja se može razlikovati (vidi 5.3).

Drugo postavljeno pitanje je što kada se matrica **J** ne može invertirati, odnosno nije kvadratna ili je singularna. Problem ne kvadratne Jakobijeve matrice proizlazi iz toga da stupanj slobode kinematičkog lanca (broj zglobova u lancu) može bit različit od stupnja slobode u vrhu manipulatora (obično 6-DOF, 3-DOF pozicija + 3-DOF orijentacija). Ako kinematički lanac ima više stupnjeva slobode od vrha manipulatora (engl. *underconstrained system*) tada će vjerojatno biti beskonačno mnogo redundantnih rješenja od kojih je potrebno pronaći ono najbolje. Kada kinematički lanac ima manje stupnjeva slobode nego vrh manipulatora (engl. *overconstrained system*) tada sustav vjerojatno neće imati rješenja te se je u tome slučaju poželjno što bliže približiti ciljnom položaju manipulatora. Rješavanja takvih ne kvadratnih sustava svodi se na pronalaženju pseudo-inverza **J**† te njegovog korištenja umjesto pravog inverza:

Iako je računanje preko pravog inverza i pseudo inverza daje najpreciznije rezultate u većini slučajeva, ponekad u nekim konfiguracijama Jakobijeva matrica može biti singularna pa je pronalaženje inverza nemoguće (Slika 2). Zato je umjesto inverza i pseudo inverza moguće koristiti i transponiranu Jakobijevu matricu **J**T što se u praksi pokazuje kao vrlo zadovoljavajuće rješenje kada je brzina pronalaženja rješenja važnija nego njegova preciznost. Također, korištenjem transponirane matrice nemamo problema sa singularitetima. Jedna od uobičajenih metoda gdje je potrebno zadržati preciznost je korištenje inverza i pseudo-inverza kada je to moguće, a kada se utvrdi singularitet tada se prebaci na metodu s transponiranom matricom ili CCD metodu. Određivanje singularnosti i računanje inverza/pseudo-inverza Jakobijeve matrice može se ostvariti npr. korištenjem SVD (engl. *singular value decomposition*) ili LU dekompozicije.

Slika 3 – Primjer singularne konfiguracije zglobova

**CILJ**

Još jedan važni faktor pri pronalaženju rješenja u numeričkoj metodi je određivanja koraka integracije β. Iako se može uzimati fiksni korak, preporučljivo je adaptivno ga mijenjati ili još bolje nekim postupkom optimirati (npr. postupak zlatnog reza). Prethodno naveden i korišten postupak numeričke integracije, poznat kao Eulerov postupak, može biti spor za mali korak integracije, stoga korištenje drugih metoda integracije poput klasičnog ili adaptivnog Runge-Kutta postupka može poboljšati brzinu konvergencije ka rješenju.

Nakon što smo započeli numeričku potragu za rješenjem potrebno je odrediti kriterije prekida potrage. Tri su uvjeta zaustavljanja iteriranja:

1. Kada smo pronašli zadovoljavajuće rješenje. Obično kada je rješenje, tj. vrh manipulatora u određenom tolerantnom području.
2. Zapeli smo u lokalnom minimumu. Tada je moguće ili promijeniti metodu pronalaženja rješenja (npr. prebaciti se na CCD) ili promijeniti veličinu koraka integracije ili završiti pretraživanje i vratiti najbolje pronađeno rješenje.
3. Pretraga traje vremenski predugo. Na primjer, u interaktivnom načinu rada ne bih smo željeli čekati. Tada također treba vratiti najbolje pronađeno rješenje.

Konačno, na kraju nam još jedino preostaje spomenuti rješavanje problema eksplicitnih ograničenja u zglobovima (npr. zglob se ne može rotirati za 180° nego od 0°-90°). Problem se može riješiti tako da se na kraju svake iteracije integracije vrijednosti u zglobovima koje su van ograničenja pomaknu na bližu granicu eksplicitnog ograničenja.

# Interpolacija i krivulje

Nakon što imamo definirana dva ili više ključna trenutaka objekta kojeg animiramo potrebno je omogućiti određivanje poze objekta između ključnih trenutaka, bez definiranja novog ključnog trenutka. To se može ostvariti interpolacijom na sljedeći način. Ako imamo definirano N ključnih trenutaka (F1, F2,…, FN) i svaki ključni trenutak ima definirano vrijeme početka relativno na početak animacije (T1, T2, …,TN).

1. Za vremenski trenutak t pronađemo dva ključna trenutka Fi(Ti) i Fj(Tj) između kojih se nalazi.
2. Pozu objekta u trenutku t dobijemo interpolacijom zglobova (parametara u zglobovima) definiranih u ključnim trenutcima Fi i Fj.

No, koji tip interpolacije koristiti. Najjednostavnije je koristiti linearnu interpolaciju, ali samo pomoću nje nemamo kontrolu glatkog prelaza između ključnih trenutaka pa animacija može izgledati neprirodno. Zato je dobro imati mogućnost definirati krivulju interpolacije.

# Dodatak

## Tipovi zglobova

U ovom odjeljku biti će prikazani najčešći tipovi zglobova koji se koristiti za definiranje kostura u 3-dimenzionanom prostoru, kao i njihove pripadne matrice lokalne transformacije.

**Translacijski 1-DOF zglob**

**Translacijski 3-DOF zglob**

**Rotacijski 1-DOF zglob**

**Rotacijski 2-DOF zglob**

**Rotacijski 3-DOF zglob**

TBD.

**Skalirani zglob**

TBD.

**Univerzalni 6-DOF zglob (3-DOF rotacija + 3-DOF translacija)**

TBD.

# Zaključak

Motivacija za odabir teme ovog seminarskog rada pronađena je u aplikaciji za izradu 2D animacija *Spine*. U ovome radu su teorijski obrađeni glavni aspekti suvremene računalne animacije i načini na koji se ostvaruju, poput kinematike, hijerarhijske strukture kostura i interpolacije. No, ovaj rad nije potpun opis znanja koje obuhvaća računalna animacija jer osim prethodno navedenih aspekata ona uzima u obzir i druge, poput mogućnosti definiranja i manipuliranja mrežom poligona nad objektom kojeg se animira (engl. *skinning*), definiranje fizikalno temeljenih animacija, detekcija kolizija i slično. To sve nadilazi ovaj seminarski rad i ostavljeno je za dodatnu analizu u budućnosti kao nadogradnja ovog rada.

# Literatura

**Mihajlović, Željka.** *Računalna grafika.* [Mrežno] http://www.zemris.fer.hr/predmeti/ra/predavanja/.

**Rotenberg, Steve. 2005.** CSE 169: Computer Animation. [Mrežno] 2005. http://graphics.ucsd.edu/courses/cse169\_w05/.

**Welman, Chris. 1989.** *Inverse kinematic and geometric constraints for articulated figure manipulation.* 1989.

# Sažetak

Računalna animacija danas je jedna od najvažnijih aspekata računalne grafike. Ovaj rad opisuje postupke korištene u suvremenoj računalnoj animaciji te matematiku i fiziku koja stoji iza, poput kinematike, interpolacije i krivulja. Posebno se razmatra inverzna kinematika kao alat koji olakšava proces izrade animacije. Rad je namijenjen kao teorijski okvir za proces izrade aplikacije za izradu animacija.