

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI/DIPLOMSKI RAD br. 493

**Animacija hodanja kinetičkog modela**

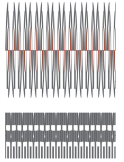
Tin Englman

Zagreb, srpanj 2012.

(prazna stranica)



**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**



ZAVRŠNI/DIPLOMSKI RAD br. 493

**Animacija hodanja kinetičkog modela**

Tin Englman

Zagreb, srpanj 2012.

(replika korica)

(izvornik zadatka)



# Sadržaj

Sadržaj .....	i
1. Uvod .....	i
2. Srodni radovi.....	3
3. Implementacija modela pokreta.....	8
4. Analiza kretnje .....	10
5. Stapanje kretnji.....	18
6. Polu-proceduralno animiranje.....	20
7. Implementacija.....	24
8. Zaključak i moguće nadogradnje .....	29
9. Literatura .....	30
Sažetak.....	31
Abstract .....	31

## 1. Uvod

U industriji računalnih igara visokokvalitetni realistični modeli s popratnim animacijama kretnje su gotovo uobičajeni. Takvi modeli odlično izgledaju u programu za modeliranje ili u vrlo specifičnom prostoru, no u promjenjivom svijetu kvaliteta im dramatično opada. Ova pojava je očita ako se promatra hodanje

čovjeka po neravnoj površini pri čemu hod izgleda neprirodno. Jedno rješenje je napraviti dodatne animacije specifično za takve situacije, no broj potrebnih animacija vrlo brzo raste za složene terene čime se opterećuje animatore te povećava cijenu projekta. Drugi pristup bi bio hodanje simulirati fizikalnim modelom. Ovaj pristup rješava problem nepravilnih terena, no, u najboljem slučaju, animatori imaju nekoliko parametra pomoću kojih je vrlo teško i ne intuitivno podešavati stil kretnje (npr. hodanje ili trčanje) a takav pristup zahtjeva mnogo računalne moći što postaje problem ako je potrebno simulirati veliki broj likova u stvarnom vremenu (što je često slučaj u računalnim igrama). U ovom radu koristi se kombinacija ova dva pristupa kako bi se iskoristila vještina animatora u kreiranju kretnji ili podatci dobiveni metodama uzorkovanja pokreta (emg. *motion capture*) da bi se modeli kretali prirodno po raznovrsnim površinama a da se pri tome očuva stil kretanja.

Postupak radi u dva koraka. Prvi korak je analiza animacije. Tijekom analize animacija se promatra u lokalnom koordinatnom sustavu za sve noge i sve animacije posebno. Na pete i palac(odnosno najjisturenije dijelove noga) postavljaju se markeri čiji se položaj uzorkuje tokom jedne iteracije animacije. Na temelju uzoraka dobiva se krivulja kretnje noge iz čega se dalje računaju ostali podatci koji se kasnije koriste animacije u stvarnom vremenu u danjim algoritmima. U drugom dijelu računaju se položaji stopala za svaki korak te korištenjem podataka izračunatih u prvom dijelu i korištenjem inverzne kinematike stvara se nova animacija koja je vjerna primjerima kretnje te također poštuje podlogu po kojoj se lik kreće.

## 2. Srodni radovi

Izrada kretnji za likove u računalnim igrama predstavlja mnoge izazove. Naviknuti smo kako se likovi kreću iz naše svakodnevice: Iz životnog iskustva znamo kako se ljudi i životinje u ponašaju, kreću i izvode razne akcije. Živa bića imaju veliki repertoar akcija koje mogu izvesti, a svaka akcija se može značajno razlikovati ovisno o kontekstu i neposrednom okruženju. Uvjerljivi likovi u igrama moraju imati tu svijest i prilagodbe u kontekstu i okoliš da izgleda kao da pripadaju u svijetu koji oni nastanjuju. U isto vrijeme svi smo razvili sposobnost iz djetinjstva da uočimo suptilnost u kretnjama koje dovode do osobnosti i emocija. Za modeliranje uvjerljivih likova, te suptilnosti gibanja moraju biti izražene kroz likove u igri. Povrh toga dolazi izazov interaktivnosti. Interaktivnost u igrama i simulacijama znači da buduće kretnje likova, ne može se uvijek odrediti unaprijed, a likovi moraju dinamički reagirati na razvoj događaja u igri, bilo pod kontrolom od strane igrača, umjetne inteligencije, ili nečeg drugog. Ove dinamičke reakcije, često se trebaju dogoditi vrlo brzo, što čini planiranje gibanja ispred vremena dodatnim izazovom.

### 2.1. Opći pristupi

Velikim brojem istraživanja razvijene su mnoge razvijanja animacija likova. Početna su istraživanja bila usmjereno na unaprijed pripremljenom (eng. *offline*) uređivanje pokreta, te njihovo kombiniranje koje pomaže animatorima stvoriti fleksibilan i novi pokret od postojećih pokretnih primjera. Dok se ovaj rad ne odnosi izravno na simulacije u stvarnom vremenu, kao što su igre, neki od općih pojmova i tehnika koriste se u tu primjenu samo u modificiranim oblicima. Za sinteze pokreta u realnom vremenu, postoji nekoliko različitih pristupa generiranja fleksibilnih i poželjnih motoričkih pokreta: Proceduralni pokret (koji se naziva algoritamski ili pokret baziran na modelu), pokret baziran na primjeru i dinamični pokret baziran na simulaciji.



Animirani likovi su važni u mnogim vrstama računalnih igara i interaktivnih simulacija. Pokreti tih likova obično se stvaraju pristupom temeljenim na primjeru prije snimljenih kretnji, bilo iz uhvaćenog ili modeliranog pokreta, koji se koriste u raznim kombinacijama. Proceduralni pristup se fokusira na stvaranje specijaliziranih tehnika koje se temelje na razumijevanju kretanja. Pristupi bazirani na primjeru koriste primjere gibanja kao ulaz i sintetiziraju nove kretnje od istih, ali također dodatno koriste proceduralne tehnike. Dinamične metode bazirane na simulaciji, izvode precizne simulacije fizičkih pokreta likova, kako bi se povećala situacijska smještenost i stvarnost u pokretu. Simulacija može biti upravljana od strane prijedloga pokreta, procesnih kontrolora, ili kombinacije prije navedenih.

## **2.2. Uređivanje unaprijed pripremljenih (*offline*) pokreta**

Mnoga istraživanja snimljenog pokreta istražila su tehnike za izmjenu podataka za dati scenarij i generalizaciju za učinkovitu ponovnu upotrebu. Te tehnike nisu izravno primjenjive na sintezi pokret prilikom izvođenja, već služe za uređivanje pokreta u fazi modeliranja. Početno je predloženo direktna interpolacija ključnim okvirima (*eng. Keyframe*), vremenskim savijanjem (*eng. time-wrapping*) i preslikavanje pomaka pokreta (*eng. motion displacement mapping*) (Bruderlin i Williams, 1995; Witkin i Popović, 1995). Gleicher je poboljšao izmjene i prilagodio nove likove održavajući željena ograničenja kao što su kontakt s okolinom uz optimizaciju putanje preko cijele sekvence (Gleicher, 1998). obraćajući se istom problemu, Lee i Shin (1999) su predstavili pristup koristeći višerazinsku B-krivulju za prilagodbu postojećeg gibanja od strane čovjekolikog lika, razdvajajući problem u izvedive podprobleme. Kovar i suradnici (2002) predstavili su metodu za suzbijanje klizanja stopala (*eng. Footskate*), baziranog na prepoznavanju i provođenju ograničenja doticaja stopala. Planiranje putanje je izvediv pristup za uređivanje pokreta, gdje su buduće akcije poznate unaprijed. Shema baziranoj na optimizaciji, prvi je predložio van de Panne (1997) za generiranje kretnji temelju navedenih otisaka stopala dvonožaca. Ta shema se proširila na kretanja četveronožaca (Torkos i van de Panne, 1998). Choi i suradnici (2003) predstavili

su metodu za planiranje prirodnog izgleda kretanje od strane dvonožca kako bi se olakšao prototip brzog gibanja i generiranje gibanja na razini zadatka. Na temelju kombinacije probabilističke planirane putanje i hijerarhijskog preslikavanja pomaka, shema daje niz prijedloga da se presele iz startnog položaja na ciljni položaj koristeći niz snimljenih isječaka. Yamane i suradnici (2004) su istraživali pristup za animiranje likova manipulirajući predmetom koji kombinira planirane putanje, ograničene na bazi inverzne kinematike. Planer puta se koristi za pronalaženje kretnje koja zadovoljava geometrijska, kinematička i ograničenja držanja, dok se zalihost u pozama likova rješava pronalaskom poza najprirodnijeg izgleda na temelju baze podataka snimljenih pokreta.

### **2.3. Čisti proceduralni pokreti**

Proceduralne metode se fokusiraju na stvaranje specijaliziranih tehnika koje se temelje na razumijevanju kretanja kodiranih u model kako ti pokreti mogu biti simulirani. Postupci mogu biti sofisticirani kao i potreba da osiguraju različite pokretne stilove ili da reagiraju na različite uvjete u simuliranom okruženju. Stupanj na kojem ova metoda uspijeva se oslanja na tome kako točno možemo shvatiti i stvoriti ljudske kretnje, ili pomicanja bilo kojeg stvorenja koje treba biti modelirano.

Bruderlin i Calvert (1993, 1996) su generirali kretanje u stvarnom vremenu simulirajući inverzno njihalo za stav nogu. Hijerarhija interaktivne kontrole usvojena je za stvaranje raznih personaliziranih ljudskih kretnji. Kai Chung i Hahn (1999) su razvili postupak za hijerarhijski sustav kontrole pokreta za generiranje ljudskog kretanja stazom na neravnom terenu. Veliki problem sa čistim proceduralnim pokretom je da ne može iskoristiti prednost vještina koje obučeni animatori imaju za stvaranje ekspresivnog pokreta, ili moć uzorkovanja pokreta (*eng. motion capture*) za stvaranje vrlo realnog pokreta. To znači da su nijanse osobina kao što su osobnost i raspoloženje teško ostvarive. Fokusi u modernim pristupima su pomaknuti uglavnom primjernim pokretima, dok je dinamički pristup (koristeći simulaciju fizike) promatram kao moderni pokret čistog proceduralnog pristupa.

## 2.4. Pokreti zasnovani na primjeru

Sinteza pokretnih primjera koristi set primjernih prijedloga s interpolacijskim shemama za izradu novih prijedloga. Primjerne metode pokušavaju pružiti niz smislenih, visoko rangiranih kontrolnih parametara za animatora ili pokretni sustav, a da pritom zadrže estetiku na primjeru pokreta u uklopljenom pokretu. Na kraju, neke primjerne metode mogu zaposliti procesne tehnike za povećanje fleksibilnosti pokreta sintetiziranog izvan formiranog prostora pokreta zasnovanih prijedlogom.

Unuma i suradnici (1995) su primjenili Fourierovu analizu na grupe podataka za interpolaciju i ekstrapoliranje ljudskog kretanja, kao što je hodanje. Wiley i Hahn (1997) primijenili su interpolacijsku tehniku na primjerne pokreta za ponovno uzimanje uzoraka podataka u mreži u parametarskom prostoru. Također koristili su linearnu interpolaciju preko jednog parametra, nagiba, kako bi generirali hodanje na nagnutoj površini. Rose sa suradnicima (1998) predstavio je tehnike raštrkane interpolacije podataka koje su pogodne za stapanje primjera pokreta koji su imali neregularne podatke u parametarskom prostoru. Interpolacija radialnih baznih funkcija i vremenskog savijanja (*eng. time-wrapping*) korišten je za generiranje gibanja s različitim emocionalnim priložima, kao što su sretna i tužna šetnja, dok je okvirna inverzna kinematika korištena za provođenje označenih ograničenja.

Sun i Metaxas (2001) koriste hibridni pristup, koji uključuje korištenje više primjerna pokreta zajedno s proceduralnim metodama koje se temelje na biomehaničkom znanju za generiranje pokreta na neravnom terenu s podesivim parametrima hoda. Koristeći svoje metode, sustav može raditi automatski, znajući kako će duljina koraka ili maksimalni otklon stopala prema van (*eng. toe-out*) utjecati na radijus okretanja, ili kako će utjecati visina i dužina koraka.

Park i sur. (2002) su predstavili metodu kretanja lika u realnom vremenu koristeći metodu stapanja pokreta na temelju podataka raspršene interpolacije. Koristeći ručno označene ključne trenutke (*eng. keytimes*), vremensko savijanje

(eng. *time-wrapping*) i inverznu kinematiku, kretanje je prilagođeno za neravan teren te također osigurava da nema klizanja stopala. Meredith i Maddock (2005) su koristili automatski identificirana ključna vremena i inverznu kinematiku ciljanjem ulaznog gibanja tijela u različitim proporcijama, kao i neravnog terena.

Sve ove metode za kretanje u realnom vremenu se temelje na korištenju nekoliko parametara, kao što su brzina i kut rotacije, a poduzimanje koraka u proizvoljnom smjeru kao i bočno ili kretanje unatrag nije podržano. Sve metode također zahtijevaju primjerne kretnje za sve kontrolirane parametre, osim metode po Parku i Sur. (2002), koji mogu prilagoditi kretnju po neravnom terenu bez potrebe za primjernim pokretima u različitim nagibima.

Forsyth (2004) je predložio metodu za kretanje likova koji predviđa budući otisak pozicije i koristi okvirnu inverznu kinematiku za podešavanje kretnje ograničenih stopalom.

Nasuprot tome, metode u ovom radu mogu generirati kretanje s koracima u proizvoljnom smjeru i na bilo kojem proizvoljnom terenu, uključujući stepenice i padine, bez potrebe za više od dva primjerna pokreta. Međutim, kada više primjernih pokreta s različitim parametrima brzine su prisutni, metoda ne koristi dodatno poboljšavanje sintetiziranih pokreta.

## **2.5. Dinamični sustavi zasnovani na simulaciji**

Veliki broj radova predstavljaju rješenja bazirana na fizici za stvaranje realne kretanje na raznovrsnom terenu. Ovi pristupi obično predstavljaju tijela likova kao zajednički povezane fizičke objekte kojima se može manipulirati koristeći zakretnu silu na pojedine zglobove. Kontroleri hoda su onda stvoreni kako bi stalno pokušavali zakretati zglobove prema svojim ciljnim kutovima koristeći proporcionalne derivatne (PD) kontrolere. To je u suprotnosti s kinematičkih pristupima, koji manipuliraju pozicijom, kutovima i brzinom bez obzira na stvorenu silu koja pokreće. Simulacija dinamike pristupa usredotočenosti na stvaranje fizičko izvedive kretanje, i kao takva, obično ne može simulirati nerealne

zahtjeve, na primjer, gdje je težište u takvom položaju da lik ne može održavati ravnotežu.

Komura i sur. (2004) su predložili tehniku za generiranje ljudskog reaktivnog pokreta reakcije na vanjske utjecaje i održavanje dinamičke ravnoteže tokom šetnje ili trčanja. Wrotek i sur. (2006) predstavili su pristup kontroliranja fizički simuliranih animiranih likova u dinamičkom virtualnom svijetu pokretima temeljenih na prijedlogu. Nerealni, ali učinkovit svijet prostora se koristi za povećanje stabilnosti i slabu oprugu za moguću ravnotežu. Iako je samo u 2D, Sok i sur. (2007) stvorili su tehniku u kojoj fizički-simulirani dvonožni likovi imitiraju ljudsko ponašanje, pretvarajući svaki primjerni pokret u fizički izvedivo, balansirano održavanje simulirane kretanje. (Yin i sur., 2007) predstavili su kontroler hoda koji može izvesti pokret od 20cm u i nagib od 6 stupnjeva na silaznim stazama. Zbog nedostatka stvarne svijesti lokalnog okoliša, navedene simulacije za veće korake i nagibe ne daju dobre rezultate te bilo koji korak koji je viši od stopala koje se podiže u ciklusu hoda vjerojatno će rezultirati neuspjehom. Korištenjem kontinuirane metode zasnovane na optimizaciji, Yin i sur. (2008) su riješili ovaj problem stvaranjem tehnike za generalizaciju kontrolera za šetnju koji je fizikalno temenjen, da bi mogli obavljati različite zadatke, kao što su penjanje na visinu od 65cm, ili guranje teških predmeta. Međutim, projektiranje pretraživačkih funkcija i odabirom odgovarajućih skupova optimizacijskih parametara, ovaj pristup zahtjeva rad korisnika i eksperimentiranje (Yin i sur., 2008, p5) za svaku novu zadaću koju lik mora biti u mogućnosti obaviti.

Osim nemogućnosti simuliranja fizički nemogućih kretanja, simulacijske metode temeljene na fizici su obično računalno skuplje od pristupa koji koriste samo kinematiku.

### **3. Implementacija modela pokreta**

Kao što je opisano u prethodnom poglavlju, čisto proceduralno gibanje, primjerno gibanje, i dinamičko simulacijsko gibanje predstavlja različite opće

pristupe problema stvaranja gibanja za kretanje lika za korištenje tijekom izvršavanja. Od navedenih primjera, ovaj je rad koristi pristup temeljen na primjerima potpomognut proceduralnim metodama.

Glavni problem s potpuno proceduralnm pristupom je nedostatak učinkovite kontrole nad stilom i osobnosti, kao i činjenica da različite vrste likova (poput životinja s različitim količinama nogu) trebaju različite modele. Budući da se metoda ovdje može primijeniti na širokom rasponu različitih igara, u potpunosti proceduralni pristup se smatra nepovoljnim.

Pristup dinamičkom pokretu koji se temelji na simulaciji fizike je dobilo popularnost u igrama, iako se još uvijek koristi samo manjini igara. Simulacijska metoda je skuplja i sama ne pruža mnogo dodatne fleksibilnosti u pokretu. Metode na temelju učenja mogu biti korištene tako da prilagode primjerne pokrete na nove situacije, ali je to trenutno komplicirani proces koji zahtijeva netrivialna ulaganja od korisnika. Također, simulacijski pristupi u pravilu isključuju stilove pokreta koji ne bi trebali biti realni. Tradicionalni primjerni pristupi su bili iznimno uspješni u generiranju visoko kvalitetne animacije pokreta, koja istinski pruža primjerske pokrete i kontrolu s više parametara. Međutim, pokret koji se kontrolira samo diskretnim skupom parametara, ne pruža fleksibilnost potrebnu za veliki skup mogućih prijedloga koji bi mogli nastati u vrlo dinamičnom okruženju. Ova teza nastoji pratiti pristup generiranog pokreta za pokretnost lika koja nadilazi neke od ograničenja s tradicionalnim primjernim pristupima, a imajući prednosti koje nadograđuje na primjerne pokrete s proceduralnim metodama.

### **3.1. Minimalni model kretanja**

Kako bi se zaobišla ograničenja metoda zasnovanim na primjerima, ovaj rad koristi metode koje nadopunjuju interpolacijske metode zasnovane na primjerima proceduralnim metodama u svrhu boljeg iskorištavanje primjera kretnji. Izazov u tome je pronaći optimalnu ravnotežu između modela koji je preopćeniti i onaj koji je previše specifičan. Model koji je previše općenit može uspjeti generirati gibanje privlačne kvalitete, ali može ostaviti mnoge probleme neriješenim. Na primjer, stapanje gibanja može se koristiti za bilo koji pokret i lik, ali često će proizvesti

greške kao što su klizanje stopala, i nadalje ostavljajući otvoren problem stapanja pravog pokreta u pravo vrijeme. Model koji je previše specifičan može ovisiti o znanju koji je točan samo za neke likove. Na primjer, većina istraživanja o proceduralnim kretanjama preuzima ljudski lik i može proizvesti odličan i realan pokret ljudskih likova, ali se ne može koristiti za bilo koji lik koji nije čovjek ili animirani pokret koji ne izgleda realistično.

Ovaj rad predstavlja model pokreta za nožna kretanja koji se može primijeniti na vrlo širok raspon likova sa nogama i stilova, a da pri tome stil kretnje bude fleksibilan.

## **4. Analiza kretnje**

U ovom dijelu opisuju se metode kojima se iz dane animacije ekstrahiraju podatci koji se kasnije koriste za proceduralno prilagođavanje. Svi podatci ovdje se računaju za svaku nogu svake dostupne animacije.

### **4.1. Uzorkovanje kretnje**

Na najjisturenije dijelove stopala(kasnije u tekstu prst i peta) pričvrste se markeri koji se prate kroz jednu iteraciju animacije. U fiksnom vremenskom intervalu (u ovom slučaju  $T = 0,005$  s) sprema se pozicija markera za sve noge, čime se dobije aproksimacija krivulje po kojoj se markeri gibaju.



**Slika 1: Uzorkovanje kretnje**

Iz ovih podataka se kasnije dobivaju ostali podatci.

## **4.2. Os gibanja**

Na temelju uzorkovanih podataka računa se sredina prsta i pete za svaki uzorak.

$$sredina(t) = \frac{prst(t) + peta(t)}{2}$$

Sredina se tako projicira na pod. Pošto se analiza vrši u referentnom koordinatnom sustavu, sredina se projicira trivijalno na ravninu  $y = 0$ . Nakon toga računa se točka  $c$  kao sredina svih uzoraka,





Slika 3: Os gibanja

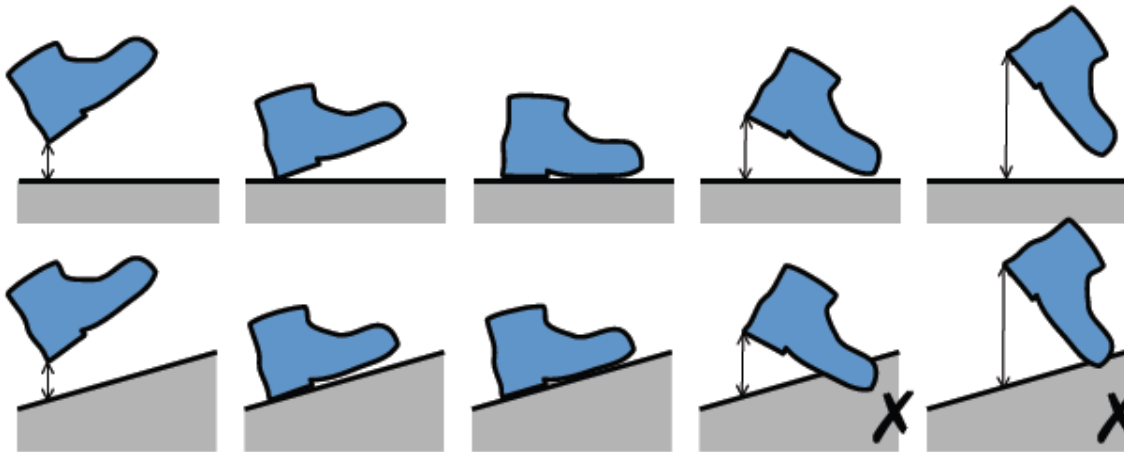
točka a koja predstavlja točku najudaljeniju od c, te točka b koja je točka najudaljenija od a.

Vektor  $b - a$  sada predstavlja procjenu osi gibanja. Ovaj podatak se sprema te se koristi u kasnijoj analizi.

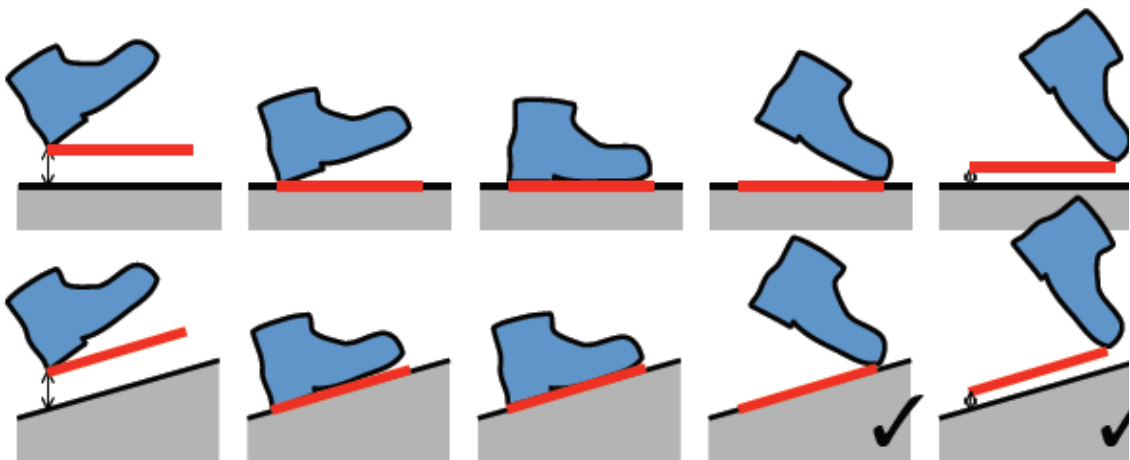
### 4.3. Trenutak ravnoteže

Trenutak ravnoteže definira se kao trenutak u gibanju u kojem je noga u najstabilnijem položaju. Računa se heuristički korištenjem nekoliko faktora. Heuristika se postiže težinskom funkcijom koja svakom uzorku dodjeljuje vrijednost koja se može interpretirati kao vjerodostojnost da je upravo ta vrijednost trenutak ravnoteže. Trenutak ravnoteže se proglašava trenutkom s najvećom vrijednošću.

$$vrijednost(t) = \alpha \max(visinaPete(t), visinaPrsta(t)) + \beta [(sredina(t) - c) \square osGibanja] \square$$



Slika 4: Nepodešeni korak



Slika 5: Podešeni korak

Funkcija uzima u obzir visinu noge kao prvi faktor te udaljenost od središnje točke kao drugi, faktori alfa i beta služe za uravnoteženje ovih faktora. Ovdje alfa je postavljen na najvišu točku svih uzoraka dok je beta dužina vektora  $b - a$ .

#### 4.4. Smjer noge

Smjer noge je vektor koji pokazuje u kojem se smjeru noga kreće za svaki uzorak projiciranom na pod. Pošto se projekcijom gubi stvarna dužina stopala, projicirani vektor potrebno je normalizirati te pomnožiti s dužinom prije projekcije.

$$d(t) = (prst(t) - peta(t)) \square pod$$

$$smjer(t) = \frac{d(t)}{\|d(t)\|} \square prst(t) - peta(t) \square$$

#### 4.5. Baza noge

Baza noge je vektor koji definira pravac po kojem se svaka noga giba u trenutku u kojem dotiče površinu gibanja. Uvijek je paralelan s površinom gibanja te se rotacija noge uvijek vrši relativno obzirom na njega kako tijekom hoda ni jedan dio noge ne prolazi kroz površinu gibanja.

Prvi korak pri određivanju baze je konstrukcija funkcije koja određuje koji dio noge je bliži podu.

$$balans(t) = atan\left(\frac{\alpha \left( visinaPeta(t) - \frac{visinaPrsta(t)}{dužinaNoge} \right)}{2\pi}\right) + 0.5$$

Arkus tangens vraća broj između  $-\pi/2$  i  $\pi/2$  te ako se rezultat te operacije podjeli s  $\pi$  te doda 0.5 dobiti će se interval od 0 do 1 što je pogodno za interpolaciju. Alfa  $>0$  je faktor koji stvara pristranost prema ekstremima 0 i 1, što je za ovaj slučaj poželjno.

Kada je poznat balans, baza noge može se izračunati sljedećom formulom

$$baza(t) = peta(t) balans(t) + (prst(t) - smijerNoge)(1 - balans(t))$$

Sada je također moguće izdvojiti bazu u trenutku ravnoteže(izračunato ranije). Dobiveni podatak se pohranjuje kao pozicija ravnoteže.

#### **4.6. Ključni trenutci**

Ključni trenutci su trenutci u ciklusu animacije koji označavaju neki bitni događaj, npr. trenutak kada noga više ne dotiče pod. Prije nego se mogu odrediti ključni trenutci potrebno je odrediti neke karakteristike krivulje gibanja.

Vertikalna komponenta pete promatra se od trenutka ravnoteže. Trenutak u kojemu se je nagib krivulje maksimalan promatrano do nekog visinskog praga. Taj trenutak se naziva odraz pete(u odsustvu prikladnijeg naziva) a predstavlja trenutak kada peta više ne dira površinu kojoj se lik hoda. U ovom slučaju prag je postavljen ja 10% maksimalne visine.

Analogno promatra se i vertikalna komponenta prsta od trenutka ravnoteže te se računa analogni podatak. Ovaj podatak se pohranjuje kao odraz prsta.

Treće se traži kraj linearne brzine. Za svako hodanje konstantne brzine mora postojati vrijeme kada je noga na podu te daje brzinu hodanju. Taj trenutak se pronalazi promatranjem akceleracije noge u svakom trenutku nakon trenutka ravnoteže te se traži nagli porast akceleracije. Mala akceleracija će uvijek postojati zbog šuma.

Analogno, no promatrajući uzorke od trenutka ravnoteže prema nazad, određuju se trenutak kada prst prvi puta dotiče pod (doticaj prsta), kada peta prvi put dotiče(doticaj pete) pod te početak linearne brzine.

Pošto postupak mora raditi za različite stilova hodanja, ne smije se pretpostaviti da je trenutak kada prsti napuštaju pod isto što i trenutak kada noga napušta pod, isto vrijedi i za petu tako da se ti pojmovi generaliziraju s trenutkom kada noga napušta pod, odnosno kada noga dotiče pod. Trenutci koji su također

zanimljivi su trenutak kada noga počinje napuštati pod, no nije još u potpunosti u zraku, te trenutak kada je noga ponovo čvrsto na zemlji.

Trenutak kada noga napušta pod uzima se kao veća (kasnija) vrijednost od trenutka kada prst napušta pod i trenutka kada peta napušta zemlju. Analogno tome trenutak kada noga dotiče zemlju uzima se kao minimum trenutka kada prst dotiče pod i trenutka kada peta dotiče pod. Trenutak kada noga počinje napuštati zemlju uzima se kao najmanja vrijednost od trenutka kada prst napušta zemlju, trenutka kada peta napušta zemlju i kraja linearne brzine. Analogno se računa i trenutak kada je noga ponovo čvrsto na zemlji. Zadnja dva podatka se koriste kako bi se izračunalo vrijeme kotrljanja noge.

U nekim stilovima hodanja moguće je da prst i peta istovremeno dotaknu pod zbog čega su uvedene dvije posebne vrijednosti: trenutak prije nego je noga potpuno na zemlji te trenutak nakon što je noga počela napuštati zemlju. Prva se računa kao manja vrijednost od trenutka kada je noga na potpuno na zemlji umanjena za minimalno vrijeme željenog trajanja kotrljanja noge i trenutka kada noga dotiče zemlju, a druga kao veća vrijednost od trenutka kada noga počinje napuštati zemlju uvećana za minimalno vrijeme željenog trajanja kotrljanja noge i trenutka kada noga napustila zemlju.

A1 – trenutak kada je prst potpuno napustio pod.

B1 – trenutak kada je peta potpuno napustila pod.

A2 – trenutak kada je prst dotakao pod.

B2 – trenutak kada je peta dotaknula pod.

C1 – kraj linearne brzine.

C2 – početak linearne brzine.

D1 – trenutak kada je noga potpuno napustila pod.

D2 – trenutak kada je noga dotakla pod.

D3 – trenutak kada je dio noge napustio pod.

D4 – trenutak kada je noga potpuno na podu.

D5 – trenutak nakon što je noga potpuno napustila pod.

D6 – trenutak prije nego je noga potpuno na podu.

e – minimalno trajanje kotrljanja noge (zadaje se proizvoljno)

$$D_1 = \max(A_1, B_1, C_1)$$

$$D_2 = \min(A_1, B_1)$$

$$D_3 = \min(A_2, B_2, C_2)$$

$$D_4 = \max(A_2, B_2)$$

$$D_5 = \max(D_3 + e, D_1)$$

$$D_6 = \min(D_4 - e, D_2)$$

#### 4.7. Smjer i dužina klizanja

S poznatim ključnim trenucima koraka moguće je izračunati smjer i dužinu klizanja noge mjerenjem puta kojega je noga prešla između trenutka u kojemu je noga dotakla pod i trenutka u kojemu je noga napustila pod te podijeliti taj vektor s vremenom između ta dva događaja.

$$smijerKlizanja = -(baza(D_1) - baza(D_2))$$

$$dužinaKlizanja = \frac{smijerKlizanja}{D_1 - D_2}$$

Smjer klizanja je negativan jer se noga, dok je na podu, uvijek kreće u smjeru suprotnom od smjera gibanja.

#### 4.8. Određivanje svojstva ciklusa

S poznatim vrijednostima smjerova i dužine klizanja za svaku nogu, moguće je izračunati ukupni smjer i put koji se pređe u jednom ciklusu animacije uzimanjem prosjeka dužine i smjera klizanja svih nogu.

$$putCiklusa = \frac{\sum dužinaKlizanja}{brojNogu}$$

$$smijerCiklusa = \frac{\sum smijerKlizanja}{brojNogu}$$

$$brzinaCiklusa = \frac{putCiklusa}{trajanjeCiklusa}$$

Iako prosjek nije sofisticiran, u praksi se pokazao vrlo efikasan. Prosjek ne bi dao dobar rezultat za animacije u kojima se lik ne kreće konstantnom brzinom, no takve animacije se ne promatraju u ovom radu. Za animacije lošije kvalitete, gdje jedna noga je malo brža od druge, greška će se rasporediti na način da će jedna noga malo brzati, dok će druga malo kasniti, što je gotovo neprimjetno za većinu animacija.

#### 4.9. Normalizacija trajektorije baze noge

U ovom trenutku poznata je trajektorija baze noge u prostoru lika. Zbog praktičnosti uporabe u vremenu izvođenja trajektoriju je potrebno transformirati u normaliziranu formu tako da noge uvijek kližu u smjeru z osi od (0,0,0) do (0,0,1). Prvo se pronađe trajektorija u globalnom prostoru na način da se prati baza noge relativno na referentnu točku koja se giba u obrnutom smjeru od smjera ciklusa putem ciklusa u vremenu trajanja ciklusa. Ta se trajektorija zarotira rotacijom Q koja ju usmjeri u smjer z osi. Z komponenta tako dobivene trajektorije dijeli se sa putem ciklusa čime se svodi na dužinu 1.

#### **4.10. Sažetak analize kretanja**

Ciklusi hodanja i trčanja su ciklusi kretnje gdje se lik kreće na mjestu na zamišljenoj ravnini konstantnom brzinom. Lik se kreće po horizontalnoj ravnini i jedan ciklus čini točno jedan korak.

Za svaki korak ekstrahira se određeni broj svojstva koja se kasnije koristi u vremenu izvršavanja.

Analiza se primarno zasniva na promatranju trajektorije prsta i pete iz čega se dalje računaju ostale vrijednosti. Na kraju analize svaka noga sadrži informacije o ključnim vremenima, kao što su trenutak ravnoteže, trenutak kada noga napušta te ponovo dotiče pod, kao i vrijeme kada je noga potpuno na zemlji. Na temelju ovih podataka izračuna se veličina i brzina svakog koraka čime je definirana kretanja cijelog lika. Podatci su na kraju spremljeni u normalnoj formi kako bi se lako koristili prilikom izvođenja u stvarnom vremenu.

### **5. Stapanje kretnji**

Stapanje kretnji je postupak koji omogućuje da kosti lika ne pokreće samo jedan primjer kretnji već da istovremeno djeluje više primjera kretnji koristeći težinski prosjek. U ovom poglavlju raspravlja se o postupku koji omogućava da se stil animacije kontrolira stilističkim parametrima dok se fizikalni parametri podešavaju automatski. Uz to pokazuju se metode sinkronizacije animacija i prostorne distorzije (*eng. motion wrapping*)

#### **5.1. Kontrola parametara i prijelaza**

Primjeri kretnji mogu varirati na više načina. Brzina i smjer su dvije dimenzije u kojima primjeri kretnje mogu imati drugačije parametre. Dok su brzina i smjer mjerljivi parametri, drugi parametri kao sreća ili uzbuđenost, su stilistički parametri te se ne mogu automatizirano mjeriti. Ove parametre naime mogu animatori i dizajneri postaviti automatski. S velikim brojem primjera kretnji s različitim

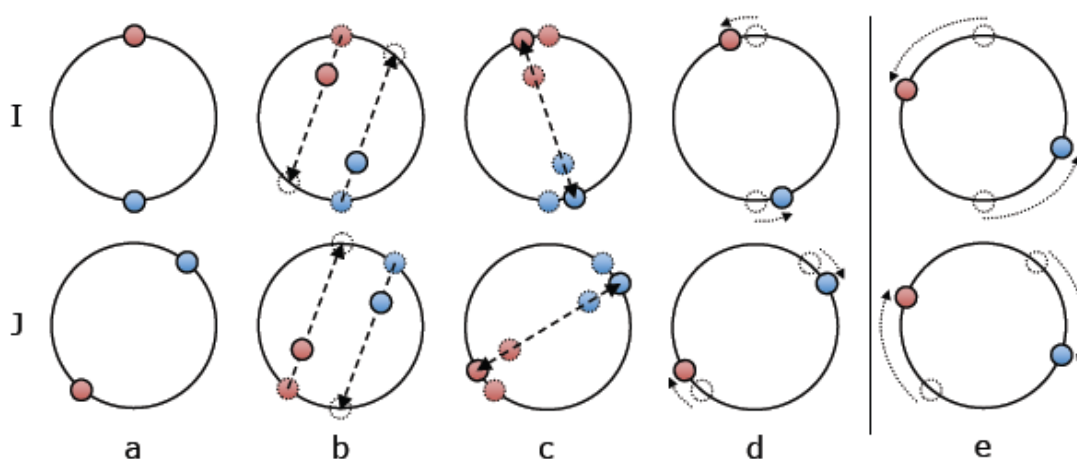


stilističkim i fizikalnim parametrima postupak dodjeljivanja odgovarajućih težinskih parametara može postati vrlo kompleksan. Automatizirane metode mogu podesiti većinu parametara te dati korisniku kontrolu parametara više razine kako bi se postigao željeni učinak. U ostatku rada fokusira se samo na stapanje fizikalnih parametara.

## 5.2. Centriranje prijelaza

U ovom radu koristi se metoda koja centrira trenutke ravnoteže tako da što je više moguće podudaraju za različite cikluse. Ovim se postiže da se animatori ne moraju brinuti da su ciklusi u različitim primjerima kretnje sinkronizirani. Npr. primjer kretnje za trčanje može početi lijevom nogom dok primjer kretnje za hodanje može početi desnom nogom, sustav će osigurati da su sinkronizirani prije stapanja.

Centriranje se postiže tako da se pohrani vremenski pomak od 0 do 1 za svaku kretnju. Svaki primjer kretnje tada se uspoređuje sa svakim drugim. Algoritam centriranja iterativno primiče trenutke ravnoteže nogu sve dok se ne centriraju, kao što je ilustrirano na slici.



Slika 6: Centriranje ciklusa. (a) Inicijalni položaji za ciklus I i J su različiti. (b) Trenutci ravnoteže pretvoreni su u 2D vektore te se primiču trenutcima ravnoteže druge animacije. (c) 2D vektori se vraćaju nazad u vremenske trenutke. (d) Nakon jedne iteracije trenutci ravnoteže su se približili. (e) Nakon više iteracija trenutci ravnoteže su gotovo potpuno poravnati.

## 6. Polu-proceduralno animiranje

U prethodim poglavljima opisane su metode za analizu određenih svojstava primjera kretnji. U ovom poglavlju predstavljaju se metode koje procjenjuju gdje će noga biti u sljedećem koraku te kako podesiti kosti lika da ta kretnja izgleda što je prirodnije moguće, a da se pri tome ostane vjerno stilu primjera kretnje.

### 6.1. Svojstva koraka

Na temelju ranije izračunatih svojstava definiraju se faze u koraku koje se kasnije koriste.

**Ciklus noge:** Vrijednost koja se linearno povećava od 0 do 1 počevši od trenutka ravnoteže sve do ponovnog trenutka ravnoteže.

**Vrijeme klizanja:** Vrijednost koja se linearno povećava od 0 do 1 počevši od trena kada je noga i potpunosti na podu do trenutka kada se noga počinje dizati s poda.

**Vrijeme leta:** Vrijednost koja se linearno poveća od 0 do 1 počevši od trenutka kada je noga napustila pod do trenutka kada je noga ponovno na podu.

**Otisak noge:** Mjesto na podu gdje se u nekome trenutku nalazi stopalo.

**Korak:** Korak od jednog otiska noge do drugog.

### 6.2. Pozicioniranje noge na zemlji

Prilikom hodanja, bili toga svjesni ili ne, stalno gledamo na pod procjenjujući sljedeći korak. Dok hodamo uzbrdo podižemo noge više kako ih ne bi zamahnuli u pod te kad hodamo preko prepreke, koliko god mala bila, podižemo mogu kako

ne bi zapeli. Dobar primjer ovoga je što svi vrlo lako hodamo niz stepenice osim kada nosimo veliku objekt koja nam blokira pogled, upravo zbog toga što ne možemo procijeniti sljedeći korak.

### 6.3. **Predviđanje otiska noge**

Trenutak ravnoteže je odabran kao referenca prema kojoj se predviđa otisak noge. Trenutak ravnoteže može se dogoditi u bilo kojem trenutku unutar vremena klizanja, no u najčešće se nalazi negdje oko sredine, time se dobiva da će se fizikalni trenutak ravnoteže dobro poravnati s onim koji su predvidjeli animatori, što je intuitivno poželjno.

Kada lik promjeni brzinu hodanja tijekom koraka, predviđeni sljedeći korak mora se također promijeniti stoga se pozicija sljedećeg koraka kontinuirano procjenjuje sve dok noga ne dotakne pod i završi svoje vrijeme leta, pri čemu predviđeni korak postaje trenutni te se počinje procjenjivati sljedeći korak. Trenutni korak naravno se ne mijenja tijekom hoda, pretpostavka je da se lik ne može poskliznuti.

Trenutak kada će noga doseći otisak noge pronalazi se množenjem trajanja ciklusa s ostatkom trajanja ciklusa noge dodano na trenutno proteklo vrijeme.

$$trenutakSljedećegKoraka = vrijeme + trajanjeCiklusa(1 - ciklusNoge)$$

S poznatim trenutkom doticaja, pozicija i orijentacija otiska noge dobiva se zbrajanjem trenutne pozicije lika s umnoškom trenutne brzine i ranije izračunatim trenutkom sljedećeg koraka umanjenom za proteklo vrijeme. Analogno se dobije i orijentacija.

*pozicijaSlijedećegKorakaLika = pozicijaLika + brzinaLika \* (trenutakSlijedećegKoraka - vrijeme)*

*orijentacijaSlijedećegKorakaLika = rotacijaLika + kutnaBrzinaLika \* (trenutakSlijedećegKoraka - vrijeme)*

Moguće je povećati preciznost položaja uzimajući u obzir kutnu brzinu pri procjeni sljedećeg koraka, no i ovakva implementacija daje dobre rezultate.

Dana pozicija i orijentacija izračunata je u prostoru lika, kako bi ih dobili u prostoru svijeta potrebno je napraviti sljedeće modifikacije:

$$\text{pozicijaSljedećegKorakaSvijet} = \text{pozicijaSljedećegKorakaLik} + \text{orijentacijaSljedećegKorakaLik} \square \text{pc}$$

$$\text{orijentacijaSljedećegKorakaSvijet} = \text{orijentacijaSljedećegKorakaLik} \square \text{orijentacijaRavnoteže}$$

#### 6.4. Uzemljeni otisak noge

Prethodno predviđeni otisak noge ne uzima u obzir teren na kojemu se lik kreće, potrebno ga je modificirati da bude paralelan s površinom. Kako bi se to ostvarilo traži se kolizija otiska noge s površinom na kojoj se nalazi. Radi jednostavnosti provjerava se samo kolizija pete i prsta otiska s površinom, te normale na tim mjestima. Temeljem toga otisak noge poravnava se s podlogom, iste varijable se koriste za pohranu tih podataka jer stari podatci više nisu potrebni.

#### 6.5. Trajektorija baze noge

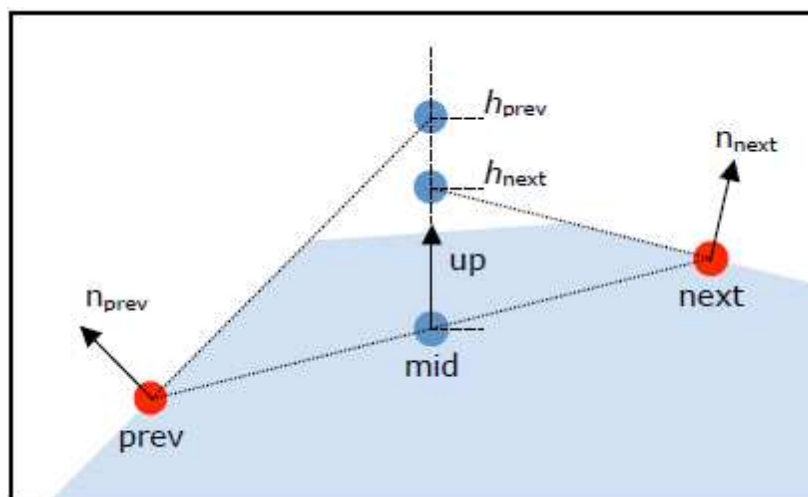
Putanja noge od trenutnog do sljedećeg utiska noge ovisi o mnogim čimbenicima. Dok dodiruje pod, noga mora biti nepomična, na mjestu gdje je predvidio sustav. Takvo ponašanje ostvaruje se inverznom kinematikom. Za putanju druge noge potrebno je uzeti u obzir nepravilnost terena, noga ne smije prolaziti kroz teren te njena kretnja mora izgledati prirodno i u skladu sa početnom animacijom.

#### 6.6. Pronalazak putanje koraka

Poznavajući poziciju, orijentaciju i trenutak u vremenu prethodnog i sljedećeg otiska noge, baza noge se mora kretati od prethodnog do sljedećeg

otiska noge tijekom vremena leta. Ova kretnja mora poštivati početnu animaciju kao i teren. Ovakvo ponašanje ostvaruje se pomoću ranije izračunate normalizirane putanje baze noge. Inicijalna putanja noge računa se interpolacijom od trenutnog do krajnjeg koraka koristeći normaliziranu putanju baze noge kao interpolacijski faktor kako bi se očuvala akceleracija i deakceleracija tokom putanje noge. Iako ovaj postupak generira realni korak, on ne uzima u obzir teren na kojem lik hoda. Potrebno je naći putanju koja s minimalno energije zaobiđe prepreke na terenu. Promatrati kompletan teren u okolini lika bi bilo računalno prezahtjevno tako da se promatra samo mjesto gdje prst i palac dodiruju pod te njihove tangente. Prvo se aritmetičkom sredinom pronađe točka između prsta i palca te se odredi vektor od te točke u smjeru koji predstavlja "gore" liku (na slici oznaka slike *up*).

$$\begin{aligned} \text{mid} &= \frac{\text{prev} + \text{next}}{2} \\ \mathbf{a} &= \text{prev} - \text{mid} \\ \mathbf{b} &= \mathbf{a} \cdot \mathbf{n}_{prev} \\ h_{prev} &= \frac{\mathbf{b}}{\mathbf{n}_{prev} \cdot \mathbf{up}} \end{aligned}$$

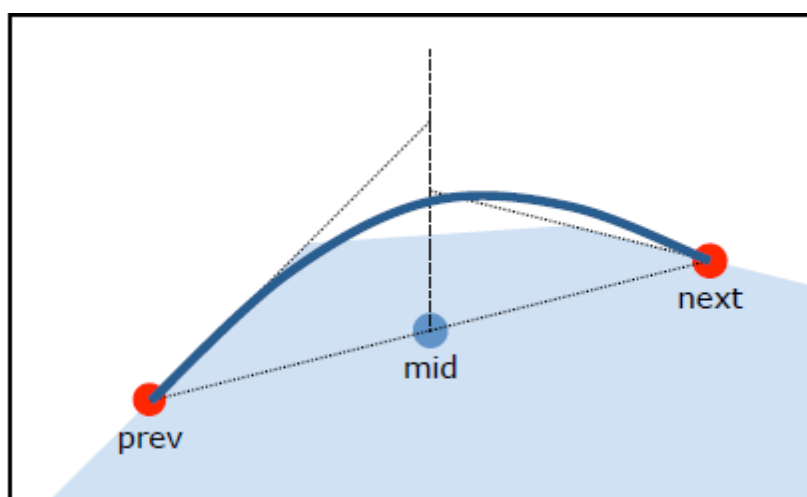


Slika 7: Analiza površine hodanja

$$h_{max} = \max(h_{prev}, h_{next})$$

$$\text{arcMagnitude} = \frac{2 \cdot h_{max}}{\pi}$$

Temeljem ovih podataka ranije izračunatoj putanji dodaje se odmak u smjeru  $up$  vektora čime se dobije krivulja koja zaobilazi prepreku.



Slika 8: Krivulja koraka





## 7. Implementacija

### 7.1. Implementacija analize kretnje

Analizom kretnje upravlja klasa *MotionAnalyser* koja sadrži sve strukture podataka te implementira sve algoritme navedene u trećem odjeljku. Klasa koristi konstrukt prijatelja klase (eng. *friend class*) s klasom *AnimatedCharacter* što joj dozvoljava pristup privatnim metodama i podacima što je nužno jer joj treba pristup unutarnjoj strukturi animacije.

#### 7.1.1.1. Uzorkovanje kretnje

Koristeći OGREov podsustav za upravljanje kosturima, na sve pete i prste manualno se postavljaju objekti.

```
mToes[0]= mCharacter->mEntity->attachObjectToBone("Joint27", ent1,  
Ogre::Quaternion::IDENTITY, Ogre::Vector3(0,0,0));
```

```
mHeels[0]= mCharacter->mEntity->attachObjectToBone("Joint27", ent2,  
Ogre::Quaternion::IDENTITY, Ogre::Vector3(0,4,25));
```

```
mToes[1]= mCharacter->mEntity->attachObjectToBone("Joint22", ent3,  
Ogre::Quaternion::IDENTITY, Ogre::Vector3(0,0,0));
```

```
mHeels[1]= mCharacter->mEntity->attachObjectToBone("Joint22", ent4,  
Ogre::Quaternion::IDENTITY, Ogre::Vector3(0,4,25));
```

mHeels i mToes su polja koja nakon ovih operacija sadrže pokazivače na objekte tipa `Ogre::TagPoint` koji daje kontrolu nad pričvršćenim objektom no nama samo treba da možemo doznati poziciju objekta. Ent1 do ent4 su objekti tipa `Ogre::Entity`.

Samo uzorkovanje se vrši tako da se animacija pokrene od početka do kraja i na dani period uzorkovanja zabilježi pozicija.

```
mCharacter->mAnimationState->addTime(T);
```

```

    for(int i = 0; i < mNumLimbs; i++) mToeSamples[i][t] = mCharacter-
>mNode->getPosition() - mToes[i]->convertLocalToWorldPosition(mToes[i]-
>getPosition());

    for(int i = 0; i < mNumLimbs; i++) mHeelSamples[i][t] = mCharacter-
>mNode->getPosition() - mHeels[i]->convertLocalToWorldPosition(mHeels[i]-
>getPosition());

```

`mToeSamples` i `mHeelSamples` su polja mapa gdje indeks `i` predstavlja redni broj noge, a `t` je trenutak u kojemu se uzorak uzima, istovremeno on predstavlja i ključ kojim se podatak dohvaća u mapi dok je `T` je period uzorkovanja.

### 7.1.1.2. Os gibanja

Prije nego se dobije os gibanja, potrebno je izračunati prosjek svih uzoraka te ih projicirati na ravninu  $y = 0$ .

```

for(int i = 0; i < mNumLimbs; i++)
    for(std::map<float,Ogre::Vector3>::iterator it =
mToeSamples[i].begin(); it != mToeSamples[i].end(); it++)
        mAverage[i][it->first] = (it->second +
mHeelSamples[i][it->first]) / 2.0f;

    for(int i = 0; i < mNumLimbs; i++)
        for(std::map<float,Ogre::Vector3>::iterator it =
mAverage[i].begin(); it != mAverage[i].end(); it++)
            mPlaneProjection[i][it->first] = Ogre::Vector3(it-
>second.x, 0, it->second.z);

```

Ovim algoritmom iterira se po svim nogama i svim uzorcima i primjenjuje formula navedena u trećem odjeljku, ovaj idiom koristi se pri računanju većine formula i te se slični algoritmi mogu lako rekonstruirati zbog čega ih nema potrebe dodatno objašnjavati.

### 7.1.1.3. Računanje ključnih trenutaka

Ključni trenutci su jedni od najbitnijih podataka koji se računaju u analizi kretnje, o njima uvelike ovisi danja kvaliteta animacije stoga ima smisla malo detaljnije analizirati algoritam njihovog pronalaska.

Pošto su algoritmi analogni, promatrat će se samo algoritam pronalaska trenutka kada prst dotiče zemlju, ostali se razlikuju samo po smjeru pretraživanja i točki koja se promatra.

Da podsjetimo, trenutak u kojemu prst dotiče pod pronalazi se tako da se promatra trajektorija prsta od trenutka ravnoteže prema nazad, odnosno u negativnom vremenskom smjeru jer je prst sigurno prvi put dotaknuo pod prije nego je noga u ravnoteži, a pri tome pamti trenutak najveće promjene visine sve dok visina prsta ne pređe neku kritičnu vrijednost.

```
float treshold = 0.1 * (maxHeight - minHeight);
float stanceHeight = mToeSamples[limb].find(mStanceTime[limb])->second.y;
```

Kritična vrijednost (*treshold*) postavlja se na 10% ukupne visine koju prst pređe tokom svoje putanje. *StanceHeight* sadrži visinu noge u trenutku ravnoteže te se promjena visine promatra relativno na nju.

```
for(std::map<float,Ogre::Vector3>::reverse_iterator it =
(std::map<float,Ogre::Vector3>::reverse_iterator)
mToeSamples[limb].find(mStanceTime[limb]); i < mToeSamples[limb].size();
it++, i++)
{
if(it == mToeSamples[limb].rend()) it = mToeSamples[limb].rbegin();
std::map<float,Ogre::Vector3>::reverse_iterator next = it;
        next++;
        if(next == mToeSamples[limb].rend()) next =
mToeSamples[limb].rbegin();
```

Idiom *reverse\_iterator* koristi se kako bi se iteriralo kroz krivulju u negativnom smjeru analogno kako se to radi i u pozitivnom. Posebna logika je implementirana kako bi iteriranje bilo cikličko, jer ne moguće da je trenutak ravnoteže u npr. trenutku 0.1 s od početka animacije, a traženi trenutka u 0.9 sekundi.

```
float diff = it->second.y - next->second.y;

if(abs(diff) > max || !i)
{
    max = diff;
    mToeStrike[limb] = it->first;
```

```

    }
    float a = it->second.y;
    float b = stanceHeight + treshold;
    if(a > b) break;
  }
}
}

```

Promjena visine jednostavno se izračuna oduzimanjem prethodne visine od trenutne te se provjerava da li je ta promjena veća od do sada najveće zabilježene. Algoritam prestaje kada je trenutna visina pete veća od visine trenutka ravnoteže uvećana za prije definirani prag.

#### 7.1.1.4. Računanje linearne komponente gibanja

Dodatna metoda za pronalazak trenutka kada noga napušta ili dotiče pod je promatranje komponente linearne brzine, glavni algoritam provodi obje metode, te za konačnu vrijednost uzima onu koju procjeni kao bolju.

```
float factor = 220.0f*T;
```

Faktor je broj koji predstavlja za koliko se mora promijeniti brzina kako bi algoritam zaključio da je kraj ciklusa linearne brzine, skaliran je s korakom uzorka kako bi bio neovisan o gustoći uzorkovanja. Određen je testiranjem

```

for(std::map<float,Ogre::Vector3>::iterator it = mFootBase[limb].find(mStanceTime[limb]); i < mFootBase[limb].size(); it++, i++)
{
  if(it == mFootBase[limb].end()) it = mFootBase[limb].begin();
  std::map<float,Ogre::Vector3>::iterator next = it;
  next++;
  if(next == mFootBase[limb].end())
    next = mFootBase[limb].begin();
}

```

Kao u prethodnom primjeru, ova logika je potrebna za cikličko iteriranje.

```
Ogre::Vector3 currentV = Ogre::Vector3(it->second.x,0,it->second.z);
Ogre::Vector3 nextV = Ogre::Vector3(next->second.x,0,next->second.z);
```

Promatra se projekcija primjera na podlogu, tako da se y komponenta postavlja na 0.

```
float currentDiff = (currentV - nextV).length();
mLinearSpeedEnd[limb] = it->first;

float currentDiffDiff = currentDiff - previousDiff;
if (firstIteration)
{
    previousDiffDiff = currentDiffDiff;
    previousDiff = currentDiff;
    firstIteration = false;
}

if(abs(currentDiffDiff) > abs(previousDiffDiff * factor)) break;
previousDiffDiff = currentDiffDiff;
previousDiff = currentDiff;
}
}
```

CurrentDiff predstavlja dužinu puta koji je baza noge prešla između dva uzorka, a currentDiffDiff razliku prijedjenih putova dviju iteracija, odnosno fizikalno predstavlja brzinu, ako sada promatramo kako se ta razlika mijenja, možemo detektirati naglu promjenu brzine, odnosno pojavljivanje akceleracije, što predstavlja kraj linearne pojave brzine.

## 8. Zaključak

Cilj ovog rada je bio stvoriti sustav za automatsko polu-proceduralno animiranje hoda kako bi se ostvarila uvjerljiva animacija. Pri tome se nastojalo iskoristiti vještine animatora kao i tehnike uzorkovanja pokreta. U drugom poglavlju dan je pregled algoritama i radova na kojima se uvelike temelji ovaj rad. Također opisani su i alternativni postupci i za koje situacije su prikladni. U četvrtom je poglavlju opisano kako se na temelju primjera kretnji računaju karakteristike koraka koje se kasnije koriste u polu-proceduralnom animiranju. U šestom poglavlju pokazuje se kako se na temelju izračunatih svojstava i analizom podloge dobije nova animacija hoda koja prati teren i poštuje stil hoda koji je zadan početnom animacijom.

U zaključku je važan vaš stav o toj problematici - je li sve riješeno, kako sadašnje rješenje zadovoljava potrebe, ima li dalje posla, u kojem smjeru bi išla daljnja istraživanja ...

## 9. Literatura

1. Johansen, R. S. (2009) , Automated Semi-Procedural Animation for Character Locomotion
2. Gleicher, M. (2001a). Comparing constraint-based motion editing methods. Graph.
3. Choi, M. G., Lee, J., and Shin, S. Y. (2003). Planning biped locomotion using motion capture data and probabilistic roadmaps. ACM Trans. Graph., 22(2):182–203.
4. Meredith, M. and Maddock, S. (2005). Adapting motion capture data using weighted real-time inverse kinematics. Comput. Entertain., 3(1):5–5.
5. Park, S. I., Shin, H. J., and Shin, S. Y. (2002). On-line locomotion generation based on motion blending.
6. Kovar, L. and Gleicher, M. (2004). Automated extraction and parameterization of motions in large data sets. Pages 559–568.

# Animacija hodanja kinetičkog modela

## Sažetak

Ovaj rad predstavlja niz tehnika za interaktivnu sintezu fleksibilnih kretnji lika. Sustav koristi primjere kretnji najčešće u obliku ključnih okvira ili uzorkovanih pokreta ciklusa hodanja i trčanja. Sustav automatski analizira svaku kretnju u fazi dizajniranja te pronalazi parametre kao što su trenutak doticanja poda ili trenutak kada noga napušta pod za svaku nogu kao i ukupnu brzinu lika. Prilikom izvođenja sustav prvo stapa kretnje temeljem trenutne brzine i trenutne rotacijske brzine zatim podešava kretnje kostiju pomoću inverzne kinematike kako bi osigurao da stopalo točno stane na podlogu. Sustav radi i za ljude i za druga bića koja se kreću pomoću bilo kojega broja nogu i bilo kojim stilom. Sustav može podesiti animaciju napravljenu proizvoljnom brzinom na ravnoj površini da radi na bilo kojoj brzini, smjeru i kutu, na bilo kojoj podlozi, uključujući stepenice i neravnine.

**Ključne riječi:** navedite ključne riječi na hrvatskom i engleskom

## Abstract

This thesis presents a framework of techniques for interactive synthesis of highly flexible character locomotion. The system uses a set of example motions primarily in the form of keyframed or motion-captured walk and run cycles. The system automatically analyzes each motion at design-time and extracts parameters such as impact and lift-off times for each foot as well as overall velocity. At runtime the system first blends the motions according to the current



velocity and rotational velocity of the character, it then adjusts the movements of the bones in the legs by means of inverse kinematics to ensure that the feet step correctly on the ground. The system works for both human and non-human characters with any amount of legs and in whatever style the provided example motions are in. It can adjust animations made for a specific speed and direction on a plain surface to any speed, direction, and curvature, on any surface, including arbitrary steps and slopes..