Vladimir Makarić

Implementacija fizike mekanih tela

U ovom radu opisan je skup algoritama i struktura podataka koji modeluju ponašanje mekanih tela pri interakciji sa čvrstom podlogom u realnom vremenu. Telo je predstavljeno skupom tačkastih masa koje međusobnim interakcijama teže da održe prvobitni oblik tela. Postignuto je vizuelno zadovoljavajuće ponašanje mekanih tela koje je vidljivo pri interakciji sa čvrstom podlogom. Implementirani su algoritmi koji pri deformaciji mekanih tela održavaju početnu površinu ili obim u svakom trenutku. Rezultati ovog rada pokazuju da je podešavanjem parametara elastičnosti moguće simulirati kretanje i ponašanje čvrstih tela pri interakciji sa podlogom bez računanja dinamike rotacije za telo kao celinu. Ovim istraživanjem nije obuhvaćema interakcija između mekanih tela, već samo interakcija mekanih tela sa čvrstim telima.

Uvod

Mekana tela su ona tela koja se prilikom interakcije sa drugim telima vidno deformišu, ali u određenoj meri teže da se vrate u svoj prvobitni oblik. Cilj ovog rada je izrada kompjuterskog modela koji oponaša takva tela prilikom sudara sa čvrstim telima. Cilj projekta nije egzaktna simulacija koja bi strogo pratila fizičke zakone, već što jednostavniji računarski model koji će grafički da prikaže interakciju mekanih tela sa okolinom. Pod računarskim modelom se podrazumeva skup algoritama i struktura podataka kojima je opisano stanje i ponašanje, u ovom slučaju, mekanih tela. Pored jednostavnosti, cilj projekta je i efikasnost, da bi se ponašanje mekanih tela moglo grafički prikazati u realnom vremenu. Pošto je jedini kriterijum valjanosti modela (pored efikasnosti) vizuelni, njegova primena ograničena je na igrice i kompjutersku animaciju. Ovaj rad se bavi dvodimenzionalnim telima, koja su predstavljena vektorski. Postoji više načina implementacije fizike mekanih tela:

- minimizacija energije (web1) (telo prelazi u oblik koji minimizuje energiju deformacije, poput balona),
- pamćenje svog originalnog oblika (web2) (svako telo pamti svoj početni oblik i usled deformacije ono pokušava da se vrati u taj oblik),
- pomoću tačkastih masa (čestica) koje su povezane oprugama koje se ponašaju po Hukovom zakonu (web3).

U ovom projektu korišćen je metod tačkastih masa.

Projekat je programiran u C++-u, za grafiku je korišćena OpenGL grafička biblioteka, za funkcionalnosti koje nisu podržane u C/C++ standardnoj biblioteci je korišćena boost biblioteka (web4).

Cilj projekta je kreiranje programa čiji je unos niz duži koje predstavljaju stranice čvrstih tela i niz tačkastih masa (čestica) koji predstavlja poligon za prikazivanje mekanog tela za koje se simulira interakcija sa čvrstim telima.

Karakteristike fizičkog sistema

Računarske simulacije funkcionišu tako što fizički sistem koji simuliraju transformišu iz stanja $S(t)$ u stanje $S'(t + \Delta t)$, gde je Δt vreme koje protekne u tranziciji $S \rightarrow S'$. U ovom projektu u fizičkom sistemu postoje samo čestice. Tela postoje samo kao interakcija između pojedinih čestica (isto kao i u svetu oko nas). Svaka čestica ima svoju masu, vektor pozicije, brzine, ubrzanja i vektor ukupne sile koja na

Vladimir Makarić (1993), Novi Sad, Železnička 39, učenik 4. razreda Gimnazije „Jovan Jovanović Zmaj” u Novom Sadu

MENTOR: Miloš Savić, Prirodno-matematički fakultet Univerziteta u Novom Sadu

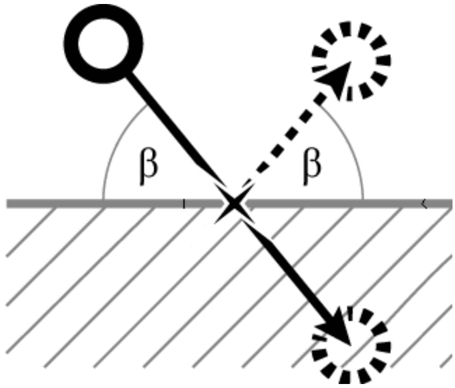
nju deluje. Da bismo transformisali sistem čestica, potrebno je za određeni interval Δt rešiti diferencijalne jednačine koje opisuju kretanje čestica (materijalnih tačaka). To radimo pomoću numeričke integracije, a u ovom projektu koristimo najjednostavniji i najneprecizniji oblik, Ojlerov. Ukoliko je Δt dovoljno malo, nepreciznost integratora je zanemarljiva što je slučaj sa ovom računarskom simulacijom ukoliko se izvršava nativno na prosečnoj mašini. U svakoj iteraciji simulacije se od vektorskog zbira svih sila koje deluju na česticu (ukupna sila) određuje ubrzanje $a = F/m$. Ubrzanje se pomnoženo sa Δt, dodaje na brzinu, a brzina se pomnožena sa Δt dodaje na trenutnu poziciju čestice. Deo programa koji vrši integraciju naziva se integrator. (U svakoj iteraciji simulacije se pored transformacije sistema radi i njegovo iscrtavanje. Vreme trajanja svake iteracije se prosleđuje integratoru sledeće iteracije kao Δt. Time se obezbeđuje nezavisnost fizičkog sistema od mašine na kojoj je program pokrenut.

Kada se nova čestica dodaje u sistem, svim njenim veličinama se zadaju proizvoljne vrednosti. Posle dodavanja u sistem, jedini način delovanja na česticu jeste preko sile koja deluje na nju, to jest njenim dodavanjem na ukupnu silu. Tako se u svakoj iteraciji na svaku česticu deluje silom gravitacije. Pošto trenutno ni jedna druga sila ne deluje na čestice one slobodno padaju.

Uvođenje čvrstih prepreka

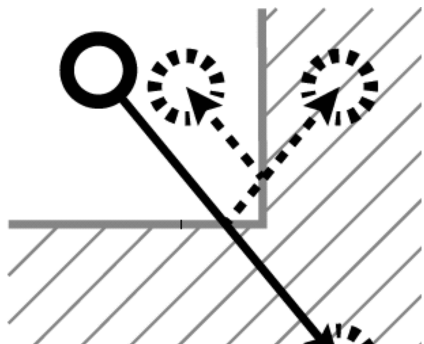
Da bi se mogle videti karakteristike mekanih tela najbolje je dovesti ih u interakciju sa čvrstim telima. Stoga su u fizički sistem dodate povezane duži koje predstavljaju statične čvrste prepreke ili tlo o koje će se čestice, a samim tim i mekana tela, odbijati. U tu svrhu je implementirana provera sudara čestice sa preprekom. Čestica nema nikakve dimenzije pa se njeno kretanje mora predvideti, inače do sudara nikad neće doći. U jednoj iteraciji čestica će se pojaviti sa druge strane duži i nastaviće dalje. Takav fenomen se naziva tunelovanje. Da bi se tunelovanje sprečilo mora se u svakoj iteraciji proveravati sudar sa duži koja predstavlja pomeraj čestice u iteraciji i dužima koje predstavljaju čvrsta tela. Duž pomeraja je duž od trenutne pozicije čestice do buduće pozicije posle integratora (pozicija+ = brzina × Δt).

Ukoliko dođe do sudara (presecanja), po zakonu refleksije se duž pomeraja reflektuje od duži čvrstog tela sa kojom je došlo do sudara. Reflektovan kraj duži je nova pozicija čestice. Reflektuje se i brzina čestice po zakonu održanja energije, da bi nastavila da se kreće kao da se fizički odbila od te duži. Slika 1 je primer ovog procesa. Crna strelica predstavlja pomeraj čestice u jednoj iteraciji, krst predstavlja sudar sa duži poda. Pošto je došlo do sudara, pozicija čestice se reflektuje. Novu reflektovanu poziciju čestice predstavlja gornji tačkasti krug. Slika 2 ilustruje moguću komplikaciju, gde se nova pozicija čestice posle refleksije nalazi u prepreci, to jest reflektovan pomeraj preseca duž prepreke.



Slika 1. Prikaz sudara čestice sa podlogom

Figure 1. Particle ground collision



Slika 2. Moguća komplikacija prilikom sudara čestice sa preprekom

Figure 2. Possible complication in the event of a particle-obstacle collision

Rešenje je rekurzivno proveravanje sudara sve dok pomeraj čestice ne preseca ni jednu duž prepreke. Na slici 2 krajnja pozicija (gornji levi tačkasti krug) se postiže posle dve iteracije algoritma, to jest posle dve refleksije. Po završetku algoritma treba proveriti da li putanju od trenutne pozicije čestice do krajnje izračunate pozicije preseca neka duž, pošto je moguće da je došlo do greške ukoliko su vrednosti brzine čestice veoma male i ukoliko se nalazi veoma blizu duži. Do greške može doći usled ograničenosti binarne reprezentacije decimalnih brojeva u računaru. Ukoliko se detektuje navedeni presek, znači da čestica leži na duži i da njenu poziciju ne treba ažurirati. Pošto je detektovanje ove greške računski zahtevno (treba opet proveravati sudare sa dužima), poziciju čestice ne ažuriramo ukoliko je brzina čestice izuzetno mala.

Opruge

Opruge predstavljaju interakciju između dve čestice, i ta interakcija teži da ih održi na određenom rastojanju. Kreiranjem mreže opruga između niza čestica koje predstavljaju omotač tela, definiše se „kostur" koji će težiti da usled interakcija tela sa podlogom održi njegov prvobitni oblik.

Opruga povezuje dve čestice i deluje silama na njih u zavisnosti od njihovog rastojanja, dužine opruge u opuštenom stanju i koeficijenta elastičnosti. Ponašanje opruge, to jest intenzitet sile koja deluje na svaku od dve čestice je definisan Hukovim zakonom, $F = -kx$ gde je x razlika opuštenog rastojanja (prirodna dužina opruge kada ni jedna sila ne deluje na nju) i trenutnog rastojanja dveju čestica koje povezuje opruga, a koje predstavlja trenutnu dužinu opruge; k je kofiecijent elastičnosti. Ova sila deluje na obe čestice u suprotnim smerovima u pravcu opruge. Pošto je simulirani fizički sistem vektorski, koristi se vektorski oblik Hukovog zakona:

$$F_{0,1} = \pm(|p_0 - p_1| - d) \cdot (p_0 - p_1) \cdot k \qquad (1)$$

gde su p_0 i p_1 pozicije krajeva opruge, odnosno pozicije čestica, d je opuštena dužina opruge, F_0 je sila koja deluje na česticu sa pozicijom p_0, a F_1 na česticu sa pozicijom p_1. Ukoliko je dužina opruge (razdaljina čestica) manja nego opuštena dužina, sile F_0 i F_1 će delovati na čestice u smeru koji će ih razdvajati (F_0 će biti u smeru od p_1 do p_0, a F_1 obrnuto). Ukoliko je dužina veća nego opuštena, sile će delovati da zbliže čestice. Delovanjem sila opruga pokušava da se vrati u svoje prvobitno opušteno stanje. Od koeficijenta elastičnosti zavisi intenzitet sila, to jest težnja opruge da se vrati u prvobitno stanje. Što je koeficijent veći to je opruga čvršća, to jest slabije se deformiše kada na nju deluju sile.

U svakoj iteraciji pre integracije za svaku oprugu se računaju sile koje se dodaju na ukupnu silu dve čestice koje opruga povezuje. Jedna čestica može biti povezana putem opruga sa više drugih čestica. Tako se formiraju *mekana tela*.

Krute opruge

Čvrsta opruga teoretski ima beskonačan koeficijent elastičnosti. U računarskoj simulaciji je nemoguće raditi sa velikim koeficijentima elastičnosti pošto kada se primenjuje sila na čestice opruge, primenjuje se na ceo interval Δt, koji uvek ima neku konkretnu vrednost, a trebalo bi da bude beskonačno mali. U prirodi za bilo koji interval Δt sila se menja beskonačno mnogo puta. Na primer ukoliko je opruga izdužena, za neki interval Δt se izračuna sila i primeni na obe čestice. Ta sila se primenjuje na ceo interval i opruga dobija veću silu nego što treba. Jer kada bi taj interval prepolovili čestica bi u prvoj polovini dobila polovinu te sile i ta sila bi promenila poziciju, tako da bi sila u drugoj polovini bila manja od sile u prvoj pošto je pozicija promenjena, to jest dužina opruge je bliža ravnotežnoj. Ovim primerom je ilustrovan problem sa kojim se susreću sve računarske simulacije fizike, jer je potrebno raditi sa beskonačno malim vrednostima, koje ne mogu ni da se predstave u računaru, a čak i da mogu, potreban bi bio beskonačno brz računar da bi mogao jednu iteraciju izvršiti za beskonačno malo vremena.

Kada se velika sila primeni na česticu (usled velikog koeficijenta elastičnosti ili velikog Δt), opruga postane nestabilna. Njena čestica dobije veliku brzinu i u sledećoj iteraciji ako nije preskočila svoj ravnotežni položaj (položaj u kome je dužina opruge jednaka opuštenoj dužini) ta brzina će još da se poveća, a ako je preskočila onda će nastaviti da „skače" gore-dole oko njega, pošto će onda dobiti veliku brzinu u suprotnom smeru. Još je situacija gora kada je više opruga povezano, jer tada se desi da usled velikog koeficijenta ili velikog Δt sistem opruga postane toliko nestabilan da „eksplodira".

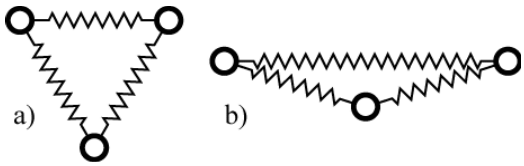
Da bi se ovo ponašanje neutralisalo i omogućilo modelovanje čvrstih opruga mora se sila primenjivati na oprugu u zavisnosti od trenutne brzine njenih krajeva (čestica). Tako se sile „ublažuju". Prvo je potrebno odrediti komponente brzine prve i druge čestice u pravcu opruge (ov_1, ov_2), pošto ta komponenta najbolje predstavlja smer i intenzitet kretanja čestice uz oprugu. Posle treba odrediti relativnu brzinu kretanja uz oprugu te dve čestice. Postoje dve takve brzine (rv_1, rv_2), istog pravca i intenziteta a različitog smera; rv_1 je brzina prve čestice (uz oprugu) sa tačke gledišta druge; rv_2 je brzina druge čestice (uz oprugu) iz tačke gledišta prve:

$$rv_1 = ov_1 - ov_2$$
$$rv_2 = -rv_1$$

Prilikom računanja sile za prvu česticu (1), F_1, na tu silu dodajemo silu koja je jednaka: $rv_2 \cdot pk$ (pk je koeficijent prigušivanja), drugoj čestici dodajemo silu istog pravca i intenziteta, samo drugog smera ($rv_1 \cdot pk$). Ukoliko bi sada opruga imala visok koeficijent elastičnosti kao i koeficijent prigušivanja, ona više ne bi poskakivala. Bila bi čvrsta, zato što ukoliko dođe do izduživanja opruge, jaka sila će delovati na česticu. Prigušivač će tu silu još povećati ukoliko se opruga trenutno izdužuje (ukoliko su brzine čestica usmerene ka spoljašnosti opruge) jer su tada F_1 i rv_1 istog smera. A posle ove iteracije kada opruga već počne velikom brzinom da se skuplja (usled jake sile koja je još pojačana prigušivačem), sila će delovati snažno, ali će prigušivač da je priguši pošto je sada rv_2 suprotnog smera od F_1. Nema potrebe još dodavati silu jer će doći do nestabilnosti. Drugim rečima prigušivač ima u vidu trenutnu brzinu čestice dok primenjuje silu na nju. Odnosno pri velikim koeficijentima elastičnosti sila nije toliko jaka da dovodi do nestabilnosti ukoliko se čestica već kreće u pravom smeru.

Povezivanje opruga

Povezivanjem čestica oprugama, stvaraju se interakcije među česticama i uz pomoć tih interakcija moguće je konstruisati tela. Najjednostavnije telo koje se može konstruisati je trougao. Prvo se tri čestice dodaju u sistem i svaka sa svakom se poveže oprugom čija opuštena dužina se podešava na rastojanje između čestica. Time se određuje opušteno stanje tela koje je formirano, to jest stanje kojem će telo težiti ukoliko dođe do deformacije. Koeficijent elastičnosti te tri opruge definiše tvrdoću trougla. Ukoliko je koeficijent mali, a masa čestica velika, opruge neće uspeti da održe formu trougla. Mora se naći balans, koji zavisi od ponašanja tela koje se želi postići. Ukoliko je željeno ponašanje elastično, deformabilno, onda se mora povećati elastičnost ili smanjiti masa, ali ne previše pošto će se dobiti čvrsto telo. Takođe je bitan ugao pod kojim su postavljene opruge, to jest uglovi trougla formiranog od čestica.



Slika 3. Primer povezivanja čestica oprugama u trougao

Figure 3. Example of a triangle formed by connecting particles with springs

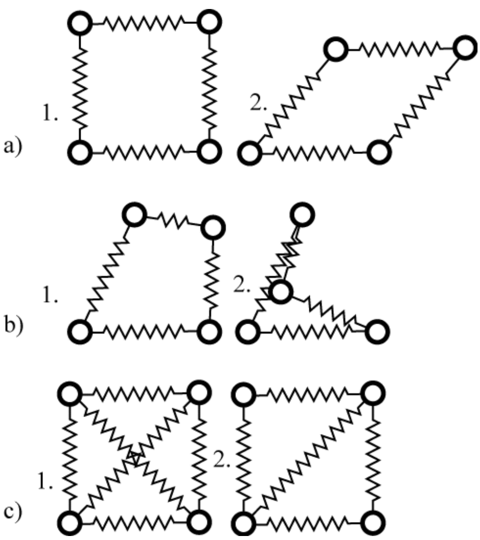
Na slici 3 se vide dva primera povezivanja opruga u trougao. U primeru a) telo (trougao) je stabilno, pošto je u pitanju jednakostraničan trougao. Stabilnost dolazi iz toga što je minimalni ugao u ovakvoj konfiguraciji maksimizovan i razdaljina između čestica je ista. Između svake dve opruge postoji minimalan ugao od 60°. U ovakvoj konfiguraciji teško može doći do situacije da jedna čestica prođe kroz drugu oprugu. Na primer, ukoliko telo sa slike 3a pada usled dejstva gravitacije i donja čestica udari o tlo, ona će dobiti brzinu ka gore, to jest ka naspramnoj opruzi. Da bi čestica mogla da prođe kroz naspramnu oprugu treba da svojim kretanjem kompresuje druge dve opruge toliko da njihova zajednička dužina bude jednaka njoj (naspramnoj opruzi). To se teško može desiti pošto će se te dve susedne opruge opirati toliko velikoj kompresiji. Sila to jest brzina bi morala biti veoma velika da bi se to desilo, stoga je ovo stabilna konfiguracija. U drugom slučaju, na slici 3b, konfiguracija je nestabilna pošto je minimalni ugao veoma mali (uglovi kod gornjih čestica). Na primer ukoliko bi telo na slici 3b udarilo o tlo, donja čestica bi morala samo malo da kompresuje susedne opruge da bi prošla kroz naspramnu oprugu, time rasturivši telo.

Opruge su odlične za modelovanje tela, pošto kroz njih sila „putuje”, dobro se propagira, baš kao u pravom telu gde sila putuje kroz atome (analogno česticama) koji su međusobno povezani raznim vezama (analogno oprugama).

Formiranje složenih tela

Ovim postupkom, algoritmom, je omogućen unos pored prostih konveksnih, veoma složenih, „izrazito” nekonveksnih tela, poput tela u obliku potkovice, u obliku plusa, zvezde i sl.

Složeno telo se definiše omotačem koji može biti konveksan ili konkavan. Omotač je uređen niz tačaka u dvodimenzionalnom prostoru. Složena tela se formiraju tako što se čestice, čije pozicije se nalaze u nizu tačaka, međusobno povezuju oprugama. Da bi formiranje tela bilo uspešno, formirano telo mora biti stabilno. Ne sme se usled interakcije sa okolinom (sudari) raspasti, to jest čestice ne smeju prolaziti kroz opruge. Složeno telo se mora sastaviti po istom principu kao i povezivanje najjednostavnijeg tela od tri čestice. Minimalni ugao mora biti maksimizovan i maksimalni unutrašnji ugao mora biti minimizovan. Sledeći korak bi bio sastavljanje četvorougla.



Slika 4. Formiranje kvadrata pomoću opruga

Figure 4. Forming a square with springs

Kada bi se omotač kod četvorougla formirao praveći analogiju sa trouglom, telo bi bilo nestabilno pošto bi pozicija čestica i površina tela mogle da se menjaju a da obim ostaje isti, to jest da opruge ostaju u opuštenom stanju. Stabilno telo pored toga što održava površinu, mora da održava i prvobitne uglove kod temena, čestica. Na slici 4 (a, b), vidimo da se prvobitno definisano telo 1 može vrlo lako deformisati u telo 2, to jest ne postoji nešto što bi ga u tome sprečilo, pošto opruge jedino održavaju dužinu. Stabilno telo uvek može da se napravi tako što bi se sve čestice povezale međusobno oprugama. To vidimo na slici 4 c1, gde je rešena nestabilnost tela pod a. Problem je što to nepotrebno usporava simulaciju, ukoliko je telo složeno, ukoliko se sastoji od većeg broja čestica. Cilj je povezati čestice sa što manje opruga, a da telo bude stabilno, odnosno da zadržava oblik definisan omotačem. Na slici 4 c2 rešen je problem nestabilnosti na efikasniji način, sa jednom dodatnom oprugom, umesto dve, time je telo podeljeno na 2 trougla. Pošto je trougao najjednostavnije telo koje može da se konstruiše pomoću najmanjeg broja tačaka (čestica), a pritom može da bude veoma stabilno ukoliko je minimalni ugao maksmizovan, logično je svako složeno telo predstaviti pomoću mreže povezanih trouglova. Drugim rečima potrebno je niz tačaka, omotač tela, to jest niz čestica triangulisati. Problem triangulacije je veoma prisutan u raznim oblastima računarstva. Razvijeni su mnogi algoritmi koji rešavaju taj problem. Algoritam koji je najpogodniji za ovu situaciju je Delanej triangulacija (web2), pošto je cilj tog algoritma baš maksimizacija minimalnog ugla svakog trougla. Delanej algoritam pravi konveksnu mrežu trouglova oko zadatog skupa tačaka, gde redosled tačaka nije bitan. Takva triangulacija je odgovarajuća kada su u pitanju konveksna tela, a u nekim slučajevima konkavnih tela nije odgovarajuća. Za potrebe ovog projekta osmišljen je posebni algoritam za generisanje mreže opruga koja će održavati oblik tela, koji pruža veći nivo kontrole nego Delanej triangulacija.

Generisanje mreže opruga

Ulaz algoritma je minimalni dozvoljeni ugao između dve opruge i omotač čestica koje definišu telo.

1. Kreira se omotač tela povezivanjem svake dve uzastopne čestice oprugom. Kreirane opruge se dodaju na listu postojećih opruga. Na slici 5a, telo je sastavljeno od čestica p0, p1,..p8. Opruge su predstavljene punom crnom duži.

2. Za česticu, izračunava se unutrašnji ugao u odnosu na omotač.

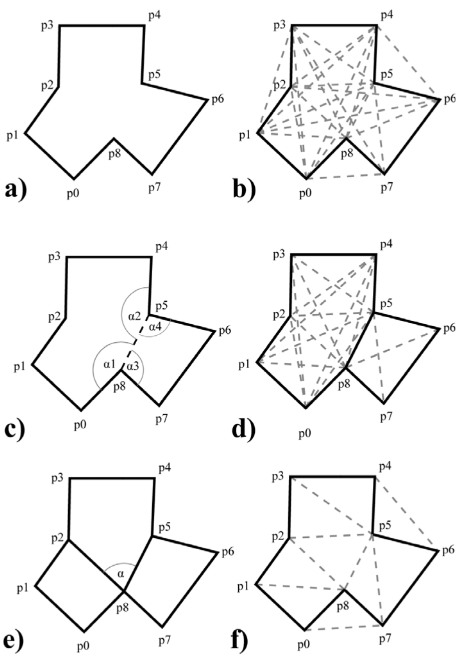
3. Određuju se sve potencijalne opruge između čestica, tako što se iz skupa svih mogućih opruga odbacuju one koje presecaju već postojeće opruge, odnosno one koje čine omotač tela. Potencijalne

opruge se vide na slici 5b. Predstavljene su isprekidanim sivim linijama.

4. Određuje se najmanji ugao koji svaka potencijalna opruga obrazuje sa ostalim postojećim oprugama. Kreira se opruga čiji je najmanji ugao veći od najmanjih uglova ostalih potencijalnih opruga. Iz toga sledi da se prvo kreiraju one opruge koje obrazuju najveće uglove između opruga. Bitno je da se takve opruge prvo dodaju pošto one uglavnom spajaju čestice koje se nalaze u središnjim delovima tela i one najviše doprinose stabilnosti. Na slici 5c, veza {p8, p5} obrazuje četiri ugla sa ostalim vezama (oprugama). Od ta četiri ugla, ugao $\alpha 4$ je minimalni ugao te potencijalne veze. A taj minimalni ugao je veći od svih ostalih minimalnih uglova koji pripadaju drugim potencijalnim vezama. Stoga će ta opruga biti dodata prva.

5. Pošto je nova veza dodata, ponovo se određuju potencijalne veze, sa istim kriterijumom da ne smeju da presecaju postojeće (slika 5d).

6. Od nove „liste" potencijalnih opruga određuje se po istom kriterijumu iz koraka 4, sledeća opruga koja će biti dodata. To je u primeru sa slike 5e veza {p2, p8} sa najvećim najmanjim uglom α.



Slika 5. Formiranje složenih tela pomoću mreže opruga

Figure 5. Forming complex bodies using a spring mesh

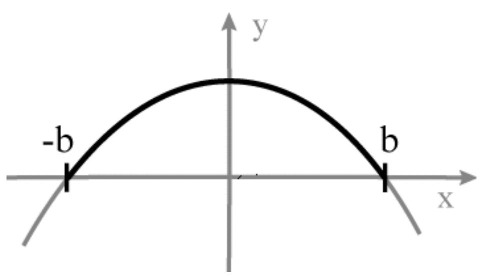
7. Postupak se nastavlja sve dok ima potencijalnih veza, ili ukoliko preostale potencijalne veze imaju minimalne uglove koji su manji od minimalnog dozvoljenog ugla između dve opruge. Slika 5f, algoritam je izvršen i kreirana je mreža opruga koja će održavati oblik tela. Unutrašnje i spoljašnje opruge su predstavljene isprekidanim sivim dužima, a opruge omotača punim crnim dužima. Jedina potencijalna veza koja nije dodata je spoljašnja veza {p1, p3}, zbog veoma malog minimalnog ugla. Treba imati u vidu da bi pri dodavanju novih opruga spoljašnje opruge trebalo da budu slabije (manji koeficijent elastičnosti) pošto one deluju samo kao suplement unutrašnjim oprugama i oprugama omotača.

Održavanje obima

Telo prilikom deformacije menja dužinu svog omotača (obima), jer neke opruge (stranice) usled delovanja raznih sila menjaju svoju dužinu. Ukoliko je potrebno da omotač tela bude konstantne dužine opruge moraju da održavaju konstantnu dužinu. Da bi se održala konstantna dužina opruge, prilikom skraćivanja opruga mora da se savije a izduživanje opruge mora da se onemogući.

Savijanje opruge

Da bi dužina opruge uvek bila ista ona mora da se na neki način savije. Savijena opruga bi u prirodi složenije delovala na čestice nego prave opruge. U ovom radu je savijanje opruge stvar kozmetike, objekat treba da izgleda kao da se savija pod pritiskom. Pošto tela dobro funkcionišu sa trenutnim modelom opruge (Hukov zakon + prigušivač) on neće biti menjan, pošto cilj projekta nije fizička korektnost. Vizuelno zadovoljavajuće savijanje opruge može se najjednostavnije predstaviti pomoću parabole.



Slika 6. Parabola koja predstavlja savijanje opruge

Figure 6. Parabola representing a bent spring

Na slici 6 prikazan je najprostiji slučaj, gde je opruga horizontalna i nalazi se na x osi. Centrirana je na y osi. Tada je jednačina parabole:

$$y(x) = a(b^2 - x^2)$$

gde je b polovina trenutne dužine opruge, a je negativni koeficijent kojim određujemo visinu parabole; b je određeno trenutnom dužinom opruge; a mora da se odredi tako da dužina luka parabole bude jednaka opuštenoj dužini opruge. To znači da pri svakoj promeni dužine opruge (ukoliko je i dalje kraća od opuštene dužine) mora da se izračuna a, tako da dužina luka parabole bude jednaka opuštenoj dužini. Za to je potrebna jednačina dužine luka parabole. Pomoću integrala je moguće izračunati dužinu luka L bilo koje funkcije f ukoliko je ona neprekidna u intervalu $[x_1, x_2]$ u kojem želimo da izračunamo dužinu luka:

$$L = \int_{x_1}^{x_2} \sqrt{1 + f'(x)^2}\,\mathrm{d}x$$

U slučaju parabole interval iznad x ose, to jest iznad opruge je $[-b, b]$. Pošto je parabola simetrična, dupla dužina nad intervalom $[0, b]$, L je opuštena dužina opruge d. $f(x)$ je $y(x) = a(b^2 - x^2)$. Tako da kada uvrstimo $y'(x) = -2ax$ i kvadriramo:

$$d = 2\int_0^b \sqrt{1 + 4a^2x^2}\,\mathrm{d}x =$$

$$= \frac{1}{2a}\left[2ab\sqrt{1 + (2ab)^2} + \sinh^{-1}(2ab)\right] \quad (2)$$

Problem sa formulom za dužinu luka parabole je što ne možemo algebarski da izrazimo a, pošto se a nalazi unutar i izvan transcedentalne funkcije $\sinh^{-1}$ (inverzni hiperbolički sinus). Stoga dužina luka mora da se aproksimira. Funkcija može da se aproksimira pomoću Tejlorovog polinoma (web6) određenog stepena. Time je izgubljena sporna transcedentalna funkcija i moguće je izraziti a u zavisnosti od opuštene dužine i trenutne dužine opruge. Problem sa Tejlorovim polinomima je što se oni ponašaju kao originalna funkcija samo u određenom intervalu. Veličina tog intervala zavisi od stepena Tejlorovog polinoma. Što je stepen polinoma veći to je izvedeni izraz za a komplikovaniji. Posle određenog stepena (zavisno od veličina b i d) izraz za a postaje dovoljno složen da standardni tipovi u C++-u nisu dovoljno veliki za njegovo računanje, zbog velikih stepena (eksponenata). Taj problem može da se reši korišćenjem biblioteka koje podržavaju velike brojeve. Aproksimacija izraza za a zavisi od količine savijanja opruge, što se više opruga savija potrebna je bolja aproksimacija (viši stepen Tejlorovog polinoma); a može da se izračuna i bez aproksimacije, tako što ćemo odrediti maksimalnu moguću vrednost za a i onda tu vrednost uvrstiti u funkciju (2) i pomoću bisekcije korigovati, sve dok rezultat funkcije ne bude dovoljno blizu d. Da bismo računali d pomoću funkcije (2), moramo da koristimo biblioteku za inverzni hiperbolički sinus, pošto u C++ standardnoj biblioteci nije podržan.

Onemogućavanje izduživanja opruge

Onemogućavanje izduživanja je realizovano uvođenjem posebnih koeficijenata elastičnosti i prigušivanja samo za slučaj kada je opruga duža od svoje opuštene dužine. Ti koeficijenti su definisani kao da je u pitanju čvrsta opruga. Ukoliko bi se opruga izdužila, posle malog broja iteracija bi se vratila na opuštenu dužinu, usled velikih sila koje bi odmah delovale na oprugu zbog povećanih koeficijenata.

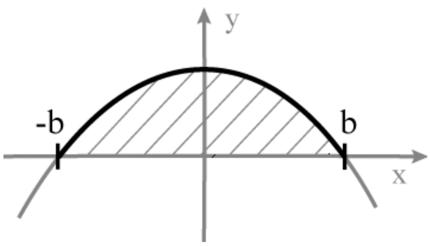
Održavanje površine

Telo prilikom deformacije menja svoju površinu, neke opruge (stranice) usled delovanja raznih sila menjaju svoju dužinu, a time se menja površina celokupnog tela. Ukoliko je potrebno da površina tela bude konstantna, promena površine mora nekako da se nadoknadi. Površina može da se nadoknadi savijanjem opruga kao kod održavanja obima, ali u ovom slučaju cilj nije konstantna dužina luka parabole, to jest dužine opruge, već se savijanjem opruga utiče na trenutnu površinu tela. Savijanjem opruge se na trenutnu površinu tela kojem ona pripada ili dodaje ili oduzima površina ispod luka parabole u zavisnosti od smera savijanja. Ukoliko se opruga savije ka unutrašnjosti tela, površina će se smanjiti, a u suprotnom ukoliko se savije ka spoljašnjosti će se povećati.

Da bi mogla da se odredi promena površine tela mora se znati početna površina P_0, to jest površina tela u originalnom, opuštenom stanju. Pošto su poznate koordinate tačaka (čestica) koje definišu telo, može se vrlo lako izračunati površina za bilo kakav oblik tog tela, bilo ono konveksno ili ne (web7).

Da bi se ustanovila promena površine tela usled moguće deformacije potrebno je u svakoj iteraciji računati trenutnu površinu tela P. Promena površine $\Delta P = P_0 - P$ predstavlja višak ili manjak površine koji se krivljenjem opruga mora neutralisati.

Savijanje opruge je kao kod održavanja konstantnog obima predstavljeno istim oblikom parabole. Svaka opruga koja je deformisana će biti savijena. Površina dobijena savijanjem opruge je površina ispod grafika parabole (slika 7, površina je šrafirana).



Slika 7. Površina dobijena savijanjem opruge

Figure 7. Surface gained by bending a spring

Opruga čija trenutna dužina je manja od opuštene dužine će uvek biti savijena ka spolja, to jest njeno savijanje će uvek povećati površinu tela. A opruga koja je izdužena će se savijati ka unutrašnjosti i njeno savijanje će smanjiti trenutnu površinu tela.

Jednačina površine ispod bilo koje funkcije koja je neprekidna na datom intervalu proizilazi iz Njutn--Lajbnicove formule. Jednostavnim računom moguće je izraziti *a* preko poludužine *b* iz jednačine za površinu.

Ako je razlika trenutne i opuštene dužine opruge apsolutna vrednost *x*, sve *x* vrednosti deformisanih opruga koje su izdužene treba sabrati u jednu promenljivu, S_i, a sve vrednosti *x* kompresovanih opruga sabrati u promenljivu S_k. Da bismo savili svaku oprugu potrebno je izračunati *a* njene parabole, a za to nam je potrebna površina pošto dužinu imamo (trenutna dužina 2*b*). Da bi površina tela bila konstantna, zbir svih površina dobijenih savijanjem svih opruga mora biti jednaka ΔP. Postupak savijanja opruga pri zahtevu da se površina održava ima dva slučaja: kada telo ima veću površinu nego što treba da ima i kada ima manju površinu. Ovde će biti opisan postupak u slučaju kada je trenutna površina tela veća nego što treba da bude, pošto je drugi slučaj veoma sličan.

Postupak

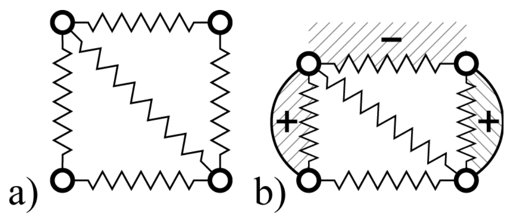
Ukoliko je ΔP negativno, znači da se površina povećala i veći broj opruga je izdužen nego što je kompresovano, odnosno $S_i > S_k$. Prema tome veći broj opruga mora da se savije ka unutrašnjosti nego ka spoljašnjosti, ali može da se desi da postoji i određen broj kompresovanih opruga (manji broj) koje moraju ka spoljašnjosti da se saviju. Iz toga sledi da izdužene opruge od tela u stvari moraju oduzeti malo veću površinu nego ΔP, a da kompresovane opruge tu razliku moraju nadoknaditi to jest pokratiti. Ta razlika je proporcionalna odnosu S_k/S_i pomnoženom sa ΔP. Površina koja određuje savijanje izdužene opruge je jednaka:

$$\frac{x}{S_k} \cdot \left(\Delta P + \frac{S_k}{S_i} \cdot \Delta P \right)$$

gde se *x* odnosi na jednu oprugu, S_k/x je deo ukupne površine ($\Delta P + (S_k/S_i) \cdot \Delta P$) koji „pripada” jednoj opruzi. Izraz za pojedinačnu površinu ispod parabole kompresovanih opruga je:

$$\frac{x}{S_i} \cdot \frac{S_k}{S_i} \cdot \Delta P$$

Na slici 8 se vidi primer održavanja površine, telo je u slobodnom padu pod a, a pri udarcu o podlogu donje čestice su odbijene u suprotnom smeru i opruge sa strane su kompresovane čime je izgubljena površina (šrafirana površina sa minusom). Površina je nadoknađena savijanjem kompresovanih opruga ka spolja (šrafirane površine sa plusom).

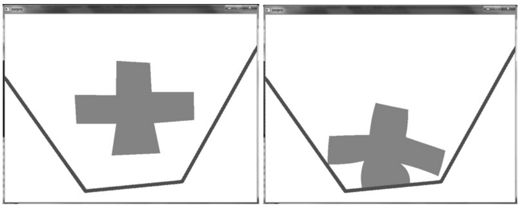


Slika 8. Primer održavanja površine

Figure 8. Example of surface retention

Zaključak

Pomoću čestica i opruga je implementiran sistem koji vizuelno zadovoljavajuće prikazuje interakciju mekih, kao i čvrstih tela sa okolinom. Sistem je veoma efikasan pošto koristi veoma jednostavnu fiziku koja se zasniva na Hukovom zakonu i ne zahteva mnogo procesorskog vremena. Postignuto je uspešno ponašanje tela pomoću jednostavnog prirodnog prin-



Slika 9. Primer simulacije

Figure 9. Simulation example

cipa čestica i njihovih interakcija. U samom sistemu ne postoje odvojena tela koja vrše svoje proračune, računaju sudare u odnosu na okolinu, ugaone brzine, momente i sl. već je sve svedeno na najjednostavnije gradivne elemente, čestice koje interaguju međusobno. Osmišljen je algoritam za konstrukciju složenih tela pomoću čestica i opruga, koji jeste računski zahtevan, ali je potrebno izvršiti ga samo jednom u toku „života" jednog tela. Predstavljen je način održavanja obima i površine tela pri deformaciji koji koristi zahtevnije proračune, ali je njegova funkcija samo suplementarna i nije deo fizičkog sistema, već samo doprinosi kozmetici. Ovaj projekat se bavio samo interakcijom između statičnih i dinamičkih tela. Da bi ovaj sistem bio kompletan, potrebno je odraditi interakciju između dva dinamička tela. Treba napomenuti da bi ovaj sistem bio nestabilan u okruženjima sa niskim FPS-om (broj frejmova-iteracija u sekundi). Po završetku rada na projektu, ovaj sistem je prekucan u actionscript-u 3 da bi mogao da se koristi u Flash igricama. Tada je primećeno da je u ovkiru Flash programa koji je ograničen na 30 fps nemoguće simulirati čvrste opruge. Problem je u Ojlerovoj integraciji koja usled nepreciznosti dodaje energiju u sistem. Rešenje za to je implementacija preciznijeg integratora. Preciznost integratora dolazi sa cenom, bolja integracija se postiže evaluacijom fizičkog sistema više puta u ovkiru jednog intervala ΔT, i nalaženjem „proseka" između dobijenih vrednosti za svaku česticu. Na primer, za RK4, Runge--Kutta integrator četvrtog stepena (web9), fizički sistem evaluira 4 puta u iteraciji što je 4 puta sporije u odnosu na Ojlerov. U ovkiru Flash okruženja je kao rešenje problema implementiran RK4 i RK2 (dva puta evaluira sistem, kompromis između Ojlerovog i RK4) integrator, gde se RK4 koristi jedino ako je potrebno simulirati veoma čvrste opruge (sa preciznošću iteratora raste čvrstoća opruga koje je moguće simulirati).

Na slici 9 vide se slike prozora simulacije u momentu kada je definisano telo u obliku plusa (levi prozor), a posle određenog broja iteracija telo je usled gravitacije udarilo u podlogu i deformisalo se (desni prozor).

Literatura

web1. http://www.cs.ucla.edu/~dt/papers/siggraph87/siggraph87.pdf

web2. http://www.beosil.com/download/MeshlessDeformations_SIG05.pdf

web3. http://en.wikipedia.org/wiki/Hooke's_law

web4. http://www.boost.org/

web5. http://en.wikipedia.org/wiki/Delaunay_trianglation

web6. http://en.wikipedia.org/wiki/Taylor_series

web7. http://www.efg2.com/Lab/Graphics/PolygonArea.htm

web8. http://nm.mathforcollege.com/topics/runge_kutta_4th_method.html

Vladimir Makarić

Implementation of Soft Body Physics

In this paper a real time soft body behavior model is described. Each body is represented by a set of point masses which are interacting with each other in order to retain the original shape of the body. A visually satisfying behavior of soft bodies was achieved, which can be best observed when the body is interacting with a solid surface. Several algorithms have been implemented which retain the original surface or circumference of soft bodies at all times, even during deformation. The results of this paper show that by modifying the elasticity parameters it is possible to simulate rigid bodies without delving into rigid body dynamics calculations.