

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 684

Fizikalno temeljen model tkanine

Hrvoje Gazibara

Zagreb, lipanj 2014.

Zahvaljujem se roditeljima i djevojci na bezrezervnoj podršci tijekom studiranja te mentorici, prof. dr. sc. Željki Mihajlović, što mi je svojim znanjem i angažiranošću kroz proteklih nekoliko godina omogućila bavljanje ovim prekrasnim područjem računarstva. Hvala i svima ostalima bez kojih mi studij jednostavno ne bi bio isti.

SADRŽAJ

Popis slika	vi
1. Uvod	1
2. Modeliranje tkanine u računalnoj grafici	2
2.1. Geometrijski postupci	2
2.2. Fizikalno utemeljeni postupci	3
2.3. Mješoviti postupci	3
3. Fizikalni model tkanine	4
3.1. Sustav masa i opruga	5
3.1.1. Model temeljen na jednodimenzijском Hookeovom zakonu	6
3.1.2. Model s geometrijskim opisom savijanja	8
3.2. Stabilizacija sustava opruga	9
3.3. Utjecaj okoline	11
3.3.1. Sila teža	11
3.3.2. Sila uzrokovana djelovanjem vjetra	11
3.4. Ukupno djelovanje sila	12
4. Numeričke metode	14
4.1. Eulerov postupak	14
4.1.1. Preciznost	16
4.1.2. Stabilnost	17
4.2. Implicitni Eulerov postupak	18
4.2.1. Preciznost	18
4.2.2. Stabilnost	19
4.3. Primjena na modelu tkanine	19

5. Vizualizacija tkanine i scene	21
5.1. Osnovni prikaz tkanine	21
5.2. Dodavanje vizualnih efekata	21
5.2.1. Primjena tekstura	21
5.2.2. Dodavanje svjetla	22
5.3. Umetanje objekata u scenu	24
5.4. Upravljanje kamerom	24
6. Otkrivanje sudara	25
6.1. Široka faza otkrivanja sudara	25
6.1.1. AABB stablo	27
6.2. Uska faza otkrivanja sudara	29
7. Razrješavanje sudara	31
7.1. Sudari tkanine s nepomičnim objektima	32
7.2. Sudari tkanine s pomičnim objektima	33
8. Programska implementacija	34
8.1. Učitavanje podataka	34
8.2. Vizualizacija rezultata	35
8.3. Izračunavanje novih stanja objekata	36
8.4. Otkrivanje sudara	37
9. Utjecaji parametara na simulaciju	38
9.1. Utjecaj na oblik i ponašanje tkanine	39
9.1.1. Svojstva opruga	40
9.1.2. Otpor zraka	41
9.2. Postupak provjeravanja parametara i implementacije	42
10. Potencijalna poboljšanja	44
10.1. Fizikalni modeli i numeričke metode	44
10.2. Otkrivanje i razrješavanje sudara	45
10.3. Sjenčanje tkanine	45
10.4. Napredniji ispitni slučajevi	46
11. Zaključak	47
Literatura	48

POPIS SLIKA

3.1. Prikaz aproksimacije tkanine sustavom čestica povezanih u mrežu	4
3.2. Jednostavni prikaz vlaka i smika	5
3.3. Različiti tipovi opruga u sustavu masa i opruga	7
3.4. Kut između dva trokuta koji definira iznos reakcijske sile uzrokovane savijanjem.	8
3.5. Ilustracija djelovanja različitih sila na jednu česticu sustava	13
4.1. Aproksimacija integrala pravokutnom formulom	16
4.2. Preciznost Eulerovog postupka za $y' = y$ i različite korake	17
5.1. Definiranje parametarskih koordinata kod uv-mapiranja	23
6.1. Određivanje kolizija metodom AABB	26
6.2. Jednostavan omeđujući volumen na primjeru spojenih piramida	27
6.3. Mogući slučajevi prilikom uske faze otkrivanja sudara.	30
8.1. Osnovna struktura razreda koji predstavljaju tkaninu	35
9.1. Primjer određivanja odnosa između piksela i metra	38
9.2. Eksplozija tkanine uzrokovana nestabilnošću numeričkoga postupka za odabrane parametre	41

1. Uvod

Usporedno s jačanjem računalne moći raste i želja za kvalitetnijim i realističnijim simulacijama i prikazima u računalnoj grafici, a vrlo jasan primjer ovih težnji moguće je pronaći u simulaciji virtualnih likova. Konstantnim razvojem postupaka u računalnoj grafici cilj je ne samo pospješiti realističnost ekspresija lica ili kretnji virtualnih likova, nego i popratnih sadržaja koji obogaćuju ostvarenu animaciju.

Nedjeljiv dio gotovo svih simulacija ponašanja virtualnih likova nedvojbeno je i tkanina jer preciznim oblikovanjem tkanine i oponašanjem svojstava stvarnih materijala moguće je značajno pospješiti realističnost prikaza te udahnuti život ostvarenim animacijama.

Kroz prethodnih tridesetak godina simulacija ponašanja tkanine predstavlja vrlo zanimljiv izazov mnogim stručnjacima te je tako razvijen širok skup metoda za simulaciju tkanine. U radu (Ng i Grimsdale, 1996) opisano je gotovo dvadeset različitih postupaka. Među navedenim postupcima većina ih se odnosi na primjenjivanje fizikalnih zakonitosti poput elastičnosti, naprežanja te deformacija i zakona očuvanja energije, uz dodatak vanjskih sila poput gravitacije, kako bi ostvareni model tkanine maksimalno odgovarao stvarnome ponašanju. U novijim radovima iz ovoga područja naglasak je na korištenju već poznatih postupaka, ali uz novije algoritme izračunavanja stanja radi ubrzavanja animacije te poboljšanjima u postupcima otkrivanja sudara te reagiranja na iste.

U svhu ostvarivanja što boljšega prikaza tkanine te imitiranja ponašanja stvarnih materijala, i u ovome radu primjenjuju se pojednostavljeni fizikalni modeli koji mogu, ovisno o svojoj složenosti, vrlo dobro preslikati ponašanje stvarne tkanine, ostavljajući animatoru da dodatnim podešavanjima prilagodi konačno ponašanje tkanine svojim zamislama.

S ciljem što bolje evaluacije opisanih postupaka za modeliranje fizikalnih modela, koristi se implementacija opisanih modela u programskome jeziku Java uz pomoć biblioteke JOGL radi korištenja standarda OpenGL u odabranom programskom jeziku.

2. Modeliranje tkanine u računalnoj grafici

Značajni interes za simuliranjem ponašanja i modeliranja tkanine u računalnoj grafici javio se 80-ih godina prošloga stoljeća, značajno kasnije od početaka u tekstilnoj industriji. Iako se u tekstilnoj industriji značajna pažnja posvećuje fizikalnim svojstvima tkanine, zbog složenosti tkanja i utjecaja različitih materijala, u računalnoj grafici težnja je na otkrivanju postupaka za postizanje što realističnijih rezultata, ali uz što manju potrošnju računalnih resursa. Samim time vrlo složena struktura tkanine pokušava se aproksimirati pojednostavljenim modelima.

U radu (Ng i Grimsdale, 1996) navedeno je devetnaest različitih metoda simuliranja tkanine, podijeljenih u tri glavne kategorije: geometrijski postupci, fizikalno utemeljeni postupci te mješoviti postupci. Iako je članak relativno star, značajan dio članaka objavljenih poslije 1996. godine zasniva se na napretcima tada opisanih postupaka, poput onih opisanih u (Baraff i Witkin, 1998), (Breen et al., 2003), (Wang et al., 2011).

2.1. Geometrijski postupci

Geometrijski postupci fokusiraju se na postizanje realističnoga izgleda tkanine ne uzimajući u obzir fizikalna svojstva same tkanine. Upravo suprotno, koriste geometrijske jednadžbe kako bi postigli uvjerljiv izgled. Jedan od prvih radova na tome području jest (Weil, 1986). U tome radu izgled tkanine modelira se korištenjem hiperbolnih krivulja, koje inače vrlo dobro opisuju oblik obješenih teških kabela.

2.2. Fizikalno utemeljeni postupci

Postupci koji se baziraju na fizikalnim procesima grade strukturu tkanine modeliranjem tkanine pravokutnim ili trokutastim rešetkama, pri čemu se na rubovima i sjecištima unutar rešetke nalaze točke (čestice) bez dimenzija, ali uz definiranu konačnu masu. Nad česticom ili pak potpunim susjedstvom čestice primjenjuju se fizikalni zakoni koji mogu biti utemeljeni na primjeni energijskih funkcija ili sila.

Prilikom korištenja funkcija koje opisuju energijska stanja računa se ukupna energija za kompletnu rešetku, a pojedine čestice raspoređuju se tako da se postigne stanje minimalne energije. U slučaju primjene sila, nad pojedinim česticama djeluju sile te se iz rezultatne sile i diferencijalnih jednadžbi dobivaju nove pozicije čestica.

Valja napomenuti da cilj nije potpuno precizno opisati sve interakcije koje se događaju unutar tkanine te samu strukturu tkanine, nego iskoristiti fizikalne zakone kako bi se dobilo ponašanje koje je vrlo slično stvarnome. Privid točne strukture jednostavnije je načiniti primjenom tekstura i sjenčanja.

2.3. Mješoviti postupci

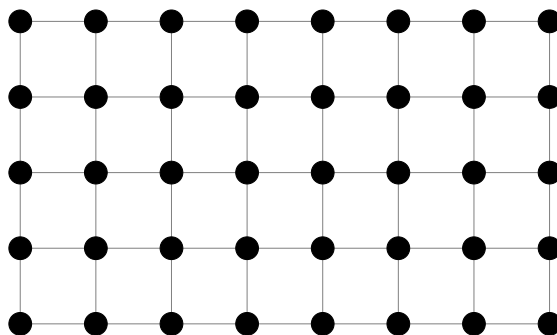
Kod mješovitih postupaka, kao što im i ime govori, radi se o kombinaciji geometrijskih i fizikalnih postupaka, najčešće s ciljem ubrzanje simulacije. Fizikalni postupci daju realističnije rezultate, ali geometrijski postupci načelno omogućavaju brže izračune pa se radi o kompromisima kako bi konačan rezultat bio uvjerljiv, ali i lako ostvariv u kontekstu brzine izvođenja. Prilikom proračunavanja konačnoga rezultata, prvo se primijeni jedan, a zatim drugi postupak.

Među prve radove u ovom području spadaju (Kunii i Gotoda, 1990) i (Rudomin, 1990). Vrlo zanimljivi recentniji rezultati dobiveni su modeliranjem tkanine doslovnom imitacijom strukture koja se dobiva tkanjem. Neki od takvih radova su (Kaldor et al., 2008), (Kaldor et al., 2010), (Yuksel et al., 2012).

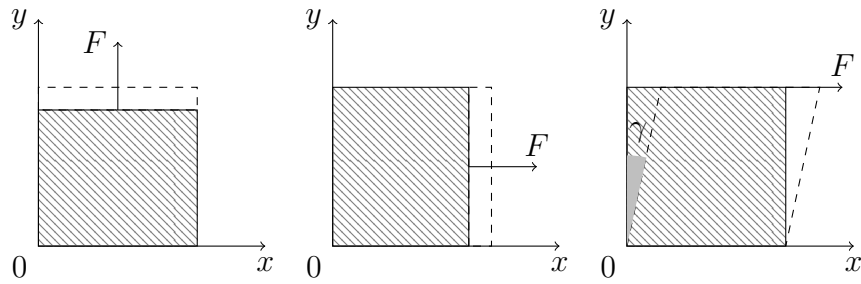
3. Fizikalni model tkanine

Fizikalni modeli tkanine iskorištavaju neke od osnovnih fizikalnih zakona kako bi što vjerodostojnije imitirali ponašanje stvarne tkanine. Nažalost, zbog kompleksnosti izrade tkanine te složenosti pojedinih materijala — niti materijala međusobno se isprepliću u različitim smjerovima — nije poznat model koji bi mogao potpuno precizno opisati međudjelovanja unutar tkanine. Samim time, svi postupci koji se koriste predstavljaju tek aproksimaciju, čiju je kvalitetu moguće procijeniti uspoređivanjem sa stvarnim ponašanjem.

Unatoč mnogim nepoznanicama, većina razvijenih postupaka simulacije tkanine iskorištava zakone elastičnosti i naprezanja radi postizanja optimalnoga ponašanja. Budući da je i struktura tkanine vrlo složena, potrebno je učiniti određene ustupke i u tome smjeru. Tako se tkanina modelira točkama koje čine rešetku sastavljenu od pravokutnika ili trokuta, kao što je prikazano na slici 3.1. Sva računanja temelje se na svojstvima točaka poput položaja, brzine i akceleracije, primjenjujući fizikalne zakonitosti na svaku od točaka pojedinačno ili uzimajući u obzir grupe točaka, ovisno o konkretnome postupku koji se koristi.



Slika 3.1: Prikaz aproksimacije tkanine sustavom čestica povezanih u mrežu



Slika 3.2: Jednostavni prikaz dvaju vlačnih naprezanja te smičnoga naprezanja. Početnih oblik prikazan je šrafiranim pravokutnikom, a deformacija isprekidanim crtama.

3.1. Sustav masa i opruga

Prilikom oblikovanja modela tkanine sustavom masa i opruga koristi se ideja oblikovanja tkanine česticama povezanima u rešetku. Svakoj čestici rešetke dodijeljena je masa, a čestice su međusobno povezane oprugama. Pomaci čestica unutar rešetke izvori su reakcijskih sila koje se javljaju kao odgovori opruga na ostvarene deformacije. Na temelju dobivenih reakcijskih sila, pomaka pojedinih čestica te prethodnoga stanja svake od čestica, moguće je izračunati novo stanje cjelokupne tkanine. Prilikom računanja novih položaja potrebno je u svakome trenutku promatrati grupe čestica jer svaka čestica spojena je oprugom s još barem jednom česticom, a elastične sile koja između njih djeluju jednakih su iznosa i smjerova, ali suprotnih orijentacija.

Neovisno o odabranome modelu tkanine, s makroskopske perspektive (s gledišta cjelokupne tkanine) moguće je uočiti postojanje nekoliko različitih tipova opterećenja koje je nužno modelirati. Položi li se tkanina na $x - y$ ravninu, tada je uočljivo da je tkaninu moguće rastezati u smjeru x -osi ili y -osi te je preklapati podizanjem dijelova tkanine s ravnine. Kada je opterećenje usmjereno u smjeru isključivo jedne od osi, tada na na cjelokupnu tkaninu djeluje **vlačno opterećenje**. Događa li se deformacija u smjeru paralelnome s ravninom, radi se o **smiku**. Smično naprezanje dovodi do nakošenja tkanine za kut γ , kao što je prikazano na slici 3.2.

Postoje dakle vlak, smik te savijanje. Mikroskopski pogled (gledište iz perspektive pojedinih čestica) otkriva da u unutrašnjosti tkanine postoje zapravo samo vlačna naprezanja između parova čestica, koja nastoje udaljiti čestice, odnosno rastegnuti oprugu koja spaja čestice. Prilikom simulacije ponašanja tkanine cilj je modeliranjem mikroskopskih utjecaja i međudnosa ostvariti prihvatljive vizualizacije rezultata makroskopskih učinaka.

Model koji simulira tkaninu ne mora koristiti jednak postupak aproksimacije za sva tri opterećenja. Moguće je koristiti opruge kao odgovor na sve tipove opterećenja ili iskoristiti neke druge metode. U prvome prezentiranom modelu koriste se opruge kako bi se tkanina oduprla svim tipovima opterećenja, dok se u drugome koristi kombinacija opruga i trigonometrijskih i geometrijskih pravila.

3.1.1. Model temeljen na jednodimenzijском Hookeovom zakonu

Realistično simuliranje ponašanja opruga moguće je ostvariti primjenjivanjem Hookeovog zakona, koji govori da je sila \vec{F} potrebna da se opruga produlji za udaljenost Δx proporcionalna iznosu elongacije. Faktor proporcionalnosti označava se slovom k , a naziva se koeficijent elastičnosti opruge. Matematički zapisano zakon glasi

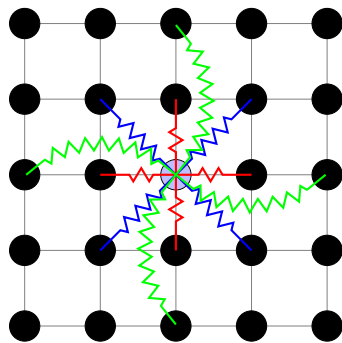
$$\mathbf{F} = k\Delta\vec{x}. \quad (3.1)$$

Kada uslijed promjene položaja čestica dođe do stezanja ili širenja opruge, javlja se reakcijska sila kojom se opruga nastoji vratiti u prvotni položaj. Magnituda te reakcijske sile jednaka je iznosu sile koja djeluje na oprugu, ali orijentacija im je drugačija. Dakle, na česticu tkanine uslijed opterećenja djeluje sila

$$\mathbf{F} = -k\Delta\mathbf{x}.$$

Kako bi se modelirao makroskopski utjecaj različitih tipova naprezanja, u model se uvode tri različita tipova opruga. **Vlačne opruge** povezuju i -tu česticu s česticama koje su neposredno lijevo, desno, gore te dolje iznad nje. Ovaj tip opruge uvodi se radi ograničavanja vlačnih naprezanja. Kako bi se ograničila smična naprezanja, uvode se **smične opruge**, koje povezuju i -tu česticu s dijagonalno susjednim česticama. U svrhu bolje manipulacije sposobnošću tkanine da se odupre savijanjima, uvode se **pregibne opruge** te one spajaju i -tu česticu s česticama koje su za dvije pozicije udaljene ulijevo, udesno te gore i dolje. Raspored opruga moguće je za jednu česticu vidjeti na slici 3.3. Crvenom bojom prikazane su vlačne opruge, plavom smične, a zelenom pregibne opruge.

Budući da je tkanina aproksimirana rešetkom, nije teško vidjeti da su svi tipovi opruga različitih duljina. Kako bi se što bolje modeliralo stvarno ponašanje tkanine, moguće je za svaki tip opruge definirati drugačiju konstantu elastičnosti, ovisno o promatranim svojstvima odabranoga materijala. Time je svaki



Slika 3.3: Izdvojena čestica s prikazom triju različitih veza koje predstavljaju različite tipove opruga

tip opruge moguće opisati parom (l_i, k_i) , pri čemu l_i označava početnu duljinu opruge, a k_i njezin koeficijent elastičnosti.

Temeljem opisanih pravila, za svaku česticu potrebno je izračunati silu prema Hookeovom zakonu toliko puta s koliko je drugih čestica spojena. Naravno, ukupan broj proračuna za svaku česticu ovisi o njezinome položaju pa će tako središnja čestica imati gotovo više nego duplo veza od čestice koja se nalazi na samome rubu rešetke. Neovisno o čestici, sila između čestica m i n može se prikazati formulom

$$\mathbf{F}_{mn} = -k_i(\|\mathbf{d}\| - l_i) \frac{\mathbf{d}}{\|\mathbf{d}\|},$$

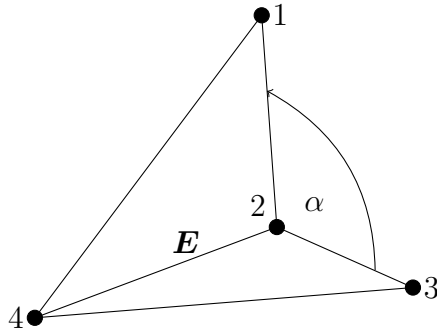
pri čemu su k_i i l_i parametri opruge između čestica m i n , a \vec{d} vektor određen radijvektorima čestica, $\vec{d} = \vec{r}_m - \vec{r}_n$. Budući da l_i predstavlja početnu duljinu opruge, a udaljenost između čestica predstavlja duljinu opruge, vidljivo je da ostatak izraza bez člana $-k_i$ označava vektor između čestica dugačak točno onoliko koliko iznosi elongacija opruge.

Proračun je dakako potrebno provesti za sve parove čestica, a ukupna sila na jednu česticu rezultat je kumulativnoga djelovanja sila elastičnosti između čestice i njezinih susjeda. Konačan izraz za računanje ukupne sile nad česticom m zbog djelovanja elastičnih sila glasi

$$\mathbf{F}_{uk,m} = \sum_{n=1}^N \left(-k_{mn}(\|\mathbf{d}_{mn}\| - l_{mn}) \frac{\mathbf{d}_{mn}}{\|\mathbf{d}_{mn}\|} \right),$$

gdje k_{mn} i l_{mn} općenito označavaju parametre opruge razapete između čestica m i n .

Nakon što je dobiven konačan izraz za silu koja djeluje na česticu, moguće je uz poznavanje mase čestice na temelju drugoga Newtonovog zakona i rješavanjem



Slika 3.4: Kut između dva trokuta koji definira iznos reakcijske sile uzrokovane savijanjem.

diferencijalne jednadžbe drugoga reda izračunati i novu poziciju čestice.

$$\ddot{\mathbf{x}} = \frac{\mathbf{F}}{m}$$

3.1.2. Model s geometrijskim opisom savijanja

U prethodnome modelu sva naprezanja opisana su korištenjem opruga. Kao mala varijacija toga modela uvodi se model u kojemu se odgovori na vlak i smik simuliraju na jednak način kao i prije, ali reakcijska sila na savijanje određuje se na temelju kuta između dvaju susjednih trokuta. Ovaj postupak istražen je, između ostaloga, u (Breen et al., 2003) i (Wang et al., 2011).

Kao što je vidljivo na slici 3.4, prilikom savijanja tkanine moguće je odrediti kut koji se javlja između dva trokuta koji dijele jedan zajednički brid. Neka je \mathbf{u}_{12} vektor određen točkama 1 i 2 na jednome trokutu, a \mathbf{u}_{32} vektor određen točkama 3 i 2 na drugome trokutu. Poznato je da je kosinus kuta između dvaju vektora jednak

$$\cos(\alpha) = \frac{\mathbf{u}_{12} \cdot \mathbf{u}_{32}}{\|\mathbf{u}_{12}\| \|\mathbf{u}_{32}\|}.$$

Kako bi sila nastala kao odgovor na savijanje bila jednaka nuli kada kut između trokuta iznosi 180° (dakle nema savijanja), a maksimalna kada su trokuti okomiti jedan na drugoga, umjesto kosinusa koristi se sinus prilikom definiranja konačnoga iznosa sile. Dodatno, što je brid što ga trokuti dijele dulji, to će sila biti veća. Konačan izraz za silu na česticu i tada je

$$\mathbf{F}_i = k \sin(\alpha/2)(h_1 + h_2)^{-1} \|\mathbf{E}\| \mathbf{u}_i. \quad (3.2)$$

Vektor \mathbf{u}_i odnosi se na vektor pozicije i -te čestice, h_1 i h_2 predstavljaju visine trokuta, a k je koeficijent koji opisuje otpor materijala na savijanje. Visine trokuta te duljina dijeljenoga brida koriste se kako bi se pokušalo modelirati anizotropno¹ ponašanje materijala.

U ovome radu ovaj model korišten je zbog sličnosti s početnim modelom te kako bi se usporedilo koliko dobro različiti modeli, ali ipak vrlo slični, mogu aproksimirati stvarno ponašanje tkanine. Osim toga, proučavanjem više jednakovrijednih modela naglašava se činjenica da egzaktno ponašanje tkanine nije potpuno poznato te da su prihvatljivi svi postupci koji mogu što kvalitetnije opisati uočena ponašanja stvarne tkanine.

3.2. Stabilizacija sustava opruga

Prvi Newtonov zakon uči nas kako svako tijelo ostaje u stanju mirovanja ili jednolikoga gibanja po pravcu sve dok vanjske sile ne uzrokuju promjenu stanja tijela. Iako je sustav masa i opruga zapravo vrlo složen sustav, i za njega općenito vrijedi slična tvrdnja. Ne primijeni li se nikada na sustav opruga vanjska sila, tada ne dolazi do elongacije opruga pa ne postoje titranja u sustavu. U slučaju da je sustav već pobuđen te opruge titraju, moguće je da se sustav, bez dodatnih primjena vanjskih sila, nalazi u stanju ravnoteže te nastavlja beskonačno titrati.

Ovo razmatranje moguće je naravno tek u teoriji jer u stvarnosti postoje sile prigušenja, poput sile otpora zraka, koje mogu dovesti do postepenoga smanjivanja titrajne frekvencije te u konačnici do stanja mirovanja sustava. Iako je potencijalnih uzroka prigušenja mnogo, u simulaciji tkanine moguće je koristiti jedinstveni model koji obuhvaća sve utjecaje, a jednostavnim namještanjem parametara moguće je odrediti utjecaj prigušenja.

U tu svrhu uvodi se dodatna sila koja djeluje unutar samoga sustava masa i opruga, a njezin iznos ovisi o brzini kretanja pojedine čestice. Budući da je za elongaciju opruge zaslužan pomak čestice, a za pomak čestice njezina brzina, definira se zasebna sila za svaku česticu kojoj je cilj smanjiti resultantnu silu koja djeluje na tijelo. Preciznije, sila djeluje u smjeru suprotnom od kretanja čestice,

¹Općenito, fizikalna i mehanička svojstva materijala mogu biti izotropna, što znači da su jednaka u svim smjerovima, ili anizotropna, što znači da se određeno svojstvo mijenja ovisno o smjeru u kojem se promatra. Metali mogu poslužiti kao primjer izotropnih materijala, a drvo je vrlo jednostavan primjer anizotropnoga materijala jer njegova svojstva ovise o tome promatraju li se u smjeru paralelome ili okomitome na godove.

smanjujući pomak čestice i njezinu brzinu u svakom sljedećem koraku. Dodatno se uvodi faktor prigušenja b kojim se može podešavati utjecaj prigušenja.

$$\mathbf{F}_{prigusenje} = -b\mathbf{v} \quad (3.3)$$

Alternativno gledano, s obzirom na to da sila prigušenja smanjuje titranje sustava, tada ona nužno mora djelovati između svih parova čestica jer na taj način definirane su opruga i sila elastičnosti nad česticama. Time se dobiva novi izraz

$$\mathbf{F}_{prigusenje} = -b(\mathbf{v}'_i - \mathbf{v}'_j), \quad (3.4)$$

u kojemu su brzine označene s \mathbf{v}'_i i \mathbf{v}'_j kako bi se naglasilo da je brzine čestica i i j nužno projicirati na pravac koji spaja čestice kako bi se dobilo djelovanje u smjeru opruge.

Izraz 3.3 pogodan je za računanje sila prigušenja koje izravno utječu na samo jednu česticu, primjerice poput sila otpora zraka. Za razliku od toga, izraz 3.4 pogodan je za izračunavanje sile prigušenja koja se javlja zbog promjene stanja više čestica, poput sile prigušenja opruge. Odabere li se tijekom implementacije i ostvarenja sila prigušenja opruge, tada je računanje sile moguće integrirati u formulu za računanje sile opruge prema Hookeovom zakonu jer u oba slučaja potrebno je u obzir uzeti dvije susjedne čestice. U tom slučaju, dobiva se formula

$$\mathbf{F}_{el} = -\left(k(\|\mathbf{d}\| - l) + b(\mathbf{v}_i - \mathbf{v}_j) \cdot \frac{\mathbf{d}}{\|\mathbf{d}\|}\right) \frac{\mathbf{d}}{\|\mathbf{d}\|}. \quad (3.5)$$

U ovome izrazu eksplicitno je navedeno preslikavanje brzina na vektor razapet česticama. Konstanta prigušenja b sada predstavlja konstantu prigušenja opruge, tako da navedeni izraz zapravo predstavlja proširenje Hookeovoga zakona za promatranu oprugu. Faktor prigušenja odabire se iz intervala $[0, 1]$, a za sve vrijednosti različite od jedan vrijedi da opruga prestaje biti savršeno elastična.

Tijekom implementacije rada isprobana su oba načina određivanja sile prigušenja, ali svaki za različitu namjenu. Formula 3.3 primijenjena je na dodavanje utjecaja otpora zraka, a formula 3.4, u svome proširenom obliku prema izrazu 3.5, korištena je kako bi se modeliralo prigušenje svih opruga u svrhu dovođenja titrajnoga sustava u ravnotežni položaj. U tom slučaju faktor prigušenja b predstavlja svojstvo opruge koja spaja čestice, jednako kao i konstanta opruge te duljina opruge.

3.3. Utjecaj okoline

S obzirom na to da je prilikom modeliranja tkanine cilj postići realistične rezultate, neminovno je da na tkaninu moraju djelovati i vanjske sile. Bez djelovanja vanjskih sila opisani sustavi bili bi statični jer ništa ne bi pobudilo opruge na titranje pa bi tkanina u svakome koraku animacije bila nepomična.

Kako vanjske sile koje djeluju na tkaninu ne bi tkaninu neprestano pomicala po sceni, neke od čestica tkanine moguće je fiksirati u prostoru. Jedna mogućnost jest proglasiti česticu bezuvjetno fiksnom, tako da resultantna sila nikada ne može pomaknuti česticu, neovisno o magnitudi. Druga mogućnost jest spojiti česticu s površinom korištenjem nove, čvrste opruge. Djelovanjem sile na česticu opruga se može malo rastegnuti, a kada iznos sile dostigne unaprijed definirani prag, veza puca i čestica postaje slobodna.

Korištenje fiksiranih točaka inherentno je važno prilikom modeliranja objekata čiji je tkanina važan dio. Bilo da se radi o odjevenome virtualnom liku, zavjesi obješenoj na kopčama ili pak zastavi privezanoj za jarbol, je omogućiti definiranje veze tkanine i drugoga objekta. potrebno

3.3.1. Sila teža

Sila koja je sama po sebi dovoljna za pokretanje titranja sustava i njegovo realistično ponašanje jest sila teža. Na svaku česticu tkanine prisutnu u rešetci potrebno je primijeniti silu težu kako bi došlo do početnih pomaka, čime se tkanina odvlači od ravnotežnoga položaja. U slučaju da niti jedna druga sila ne djeluje na sustav, sila teža rasteže tkaninu po duljini sve do stabilizacije sustava.

U slučaju primjene sile teže, na iznos resultantne sile dobivene zbrajanjem pojedinačnih elastičnih sila potrebno je još samo dodati vektor koji predstavlja smjer djelovanja sile teže pomnožen gravitacijskom konstantom, $g \approx 9.81 \frac{m}{s^2}$. Smjer djelovanja sile teže određen je smjerom koji u sceni predstavlja “dolje”, a to je najčešće vektor $\mathbf{v}_{dolje} = (0, -1, 0)$.

3.3.2. Sila uzrokovana djelovanjem vjetra

Kako bi se pojedini objekti načinjeni od tkanine realistično ponašali, uvodi se utjecaj nastao djelovanjem vjetra. U scenu se postavlja jedan ili više izvora vjetra koji su definirani vektorom koji predstavlja silu nastalu djelovanjem vjetra — \mathbf{F}_{vjetar} . Radi jednostavnosti i zbog pojednostavljenoga načina modeliranja

vjetra, moguće je utjecaj vjetra u svim dijelovima scene proglašiti konstantnim, za razliku od situacije kada pojedini objekti mogu pružati zaklon od vjetra, čime smanjuju njegov utjecaj na druge objekte.

Kako bi vjetar ipak nešto realističnije djelovao na tkaninu, utjecaj vjetra na svaku česticu korigira se faktorom koji ovisi o odnosu između normale čestice i vektora vjetra. Naime, nije isto puše li vjetar okomito na objekt ili pod nekim kutem; možemo očekivati da će vjetar imati najjači utjecaj na objekt kada puše točno okomito na njega. Korekcijski faktor moguće je modelirati skalarnim produktom između normaliziranoga vektora smjera vjetra i normalizirane normale čestice. Kada su vektor smjera vjetra i normala paralelni, tada vjetar puše točno okomito na česticu. Budući da ovaj slučaj predstavlja maksimalni utjecaj vjetra, a skalarni produkt jednak je točno jedan kada su vektori paralelni (pod uvjetom da su oba vektora normirana), ovdje skalarni produkt odgovara ideji. Kada su navedena dva vektora okomita jedan na drugoga, vjetar puše paralelno s djelićem tkanine pa treba ograničiti njegov utjecaj. U tom slučaju skalarni produkt treba biti jednak nuli, što upravo i jest slučaj za okomite vektore. U svim ostalim slučajevima korekcijski faktor bit će po apsolutnoj vrijednosti sadržan u intervalu $(0, 1)$. Predznak skalarnoga produkta uvijek je takav da prati smjer vjetra, odnosno označava puše li vjetar u smjeru normale čestice ili suprotnome.

Na temelju opisanoga postupka zaključuju se da je konačan izraz za silu uzrokovanu vjetrom na česticu i jednak

$$\mathbf{F}_{vjetar,i} = \mathbf{F}_{vjetar} \left(\frac{\mathbf{F}_{vjetar}}{\|\mathbf{F}_{vjetar}\|} \cdot \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|} \right),$$

pri čemu \mathbf{n}_i predstavlja normalu i -te čestice.

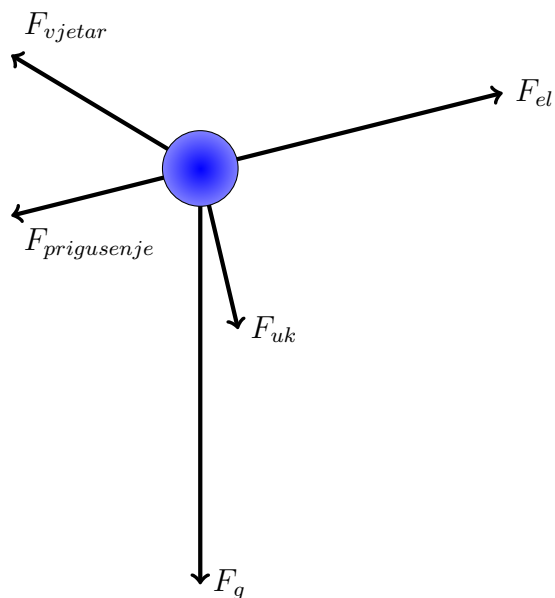
Postoji li u sceni više izvora vjetra, računa se iznos sile vjetra za svaki od izvora te se sile zbrajaju u resultantnu silu uzrokovanu svim izvorima vjetra.

3.4. Ukupno djelovanje sila

Nakon što su definirane i izračunate sve sile koje djeluju na svaku česticu sustava, potrebno je odrediti resultantnu silu nad svakom česticom. Budući da su sve sile predstavljene vektorima, računanije nije zahtjevno jer svodi se na zbrajanje vektora, što se prikazuje formulom

$$\mathbf{F}_{uk} = \mathbf{F}_g + \mathbf{F}_{el} + \mathbf{F}_{vjetar} + \mathbf{F}_{prigusenje},$$

u kojoj F_{el} predstavlja silu uzrokovanu promjenom duljine opruge radi pomicanja čestica. Slika 3.5 prikazuje jednostavan primjer djelovanja sila nad jednom izdvojenom česticom te dobivenu rezultatnu silu.



Slika 3.5: Ilustracija djelovanja različitih sila na jednu česticu sustava

4. Numeričke metode

Prilikom simulacije fizikalnih sustava i pojava na računalu javlja se značajan problem: ponašanje sustava opisano je nizom diferencijalnih jednadžbi, koje zahtijevaju korištenje postupaka integriranja radi pronalaženja egzaktnoga rješenja. Budući da se simulacija obavlja na računalu, egzaktno rješenje nije moguće pronaći jer integracija zahtjeva beskonačno finu preciznost, što na računalu nije moguće postići.

U svrhu rješavanja ovoga problema osmišljeni su različiti postupci numeričke integracije koji omogućavaju izračunavanje konačnih integrala. Na temelju tih postupaka moguće je izgraditi postupke za pronalaženje rješenja diferencijalnih jednadžbi.

Svaki od postupaka temelji se na izračunavanju rješenja u nekoliko uzastopnih točaka kako bi se došlo do krajnjega rješenja. U najjednostavnijem slučaju sve točke jednako su udaljene pa možemo govoriti o fiksnom koraku h . U složenijim slučajevima korak se mijenja tijekom provođenja postupka pa tada govorimo o adaptivnom koraku.

Budući da postupci predstavljaju tek aproksimaciju konačnoga rješenja, potrebno je obratiti značajnu pozornost na preciznost i stabilnost odabranoga postupka radi odabiranja najprikladnijega postupka za problem koji se rješava.

U nastavku rada opisana su dva postupka koje je moguće primijeniti na rješavanje diferencijalnih jednadžbi te su ukratko opisane preciznost i stabilnost za svaki od postupaka kako bi se analizirala njihova prikladnost primjene u postupku fizikalne simulacije tkanine.

4.1. Eulerov postupak

Običnu diferencijalnu jednadžbu u općenitome slučaju moguće je prikazati izrazom

$$y'(t) = f(t, y(t)), \quad (4.1)$$

pri čemu je $f(t, y(t))$ općeniti izraz za oblik prve derivacije funkcije $y(t)$. Primjerice, za funkciju $y(t) = e^{-t}$, funkcija $f(t, y(t))$ jednaka je $-e^{-t}$.

S obzirom na to da ovakva obična diferencijalna jednadžba predstavlja ponašanje sustava kroz vrijeme, potrebno je poznavati i početno stanje sustava, što se označava izrazom

$$y(t_0) = y_0, t_0 = \text{početni trenutak promatranja sustava.}$$

Obična diferencijalna jednadžba uz pripadajuće početno stanje naziva se problem s početnim uvjetima, odnosno Cauchyjev problem (Elezović et al., 2012). Na ovako opisan problem može se primijeniti numerički postupak rješavanje diferencijalne jednadžbe, a jedan takav je i Eulerov postupak.

Postupak na temelju funkcije $f(t, y(t))$ određuje vrijednost funkcije $y(t)$ bez poznavanja izraza za funkciju $y(t)$. Kompletan postupak temelji se na izračunavanju vrijednosti funkcije $y(t)$ u točki $y(t_n + h)$ na temelju već izračunate vrijednosti $y(t_n)$. Budući da je poznata vrijednost funkcije u točki t_0 , tada je vrlo lako izračunati vrijednost funkcije u točki $t_0 + h$ te svakoj sljedećoj. Ugrubo, postupak se može prikazati na sljedeći način:

$$y(t_i + h) = y(t_i) + \Delta y, \tag{4.2}$$

pri čemu je Δy izraz koji je tek potrebno odrediti.

Iz diferencijalnoga računa poznato je da se vrijednost konačnoga integrala može izračunati izrazom

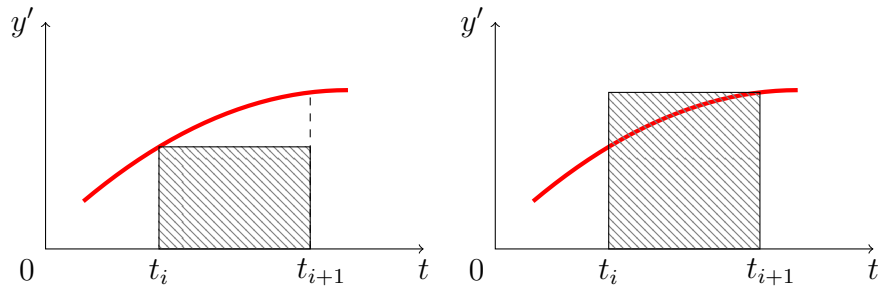
$$\int_{t_i}^{t_i+h} y'(t) dt = y(t_i + h) - y(t_i)$$

Preslagivanjem dijelova prethodnoga izraza, moguće je uvidjeti da se vrijednost funkcije u točki $y(t_i + h)$ može egzaktno dobiti računanjem izraza

$$y(t_i + h) = y(t_i) + \int_{t_i}^{t_i+h} y'(t) dt. \tag{4.3}$$

Uspoređivanjem izraza 4.2 i 4.3 uočljivo je da je Δy upravo jednako vrijednosti integrala funkcije $y'(t)$ između točaka t_i i t_{i+1} . Budući da je upravo cilj izbjegavanje izračunavanja integrala, moguće je aproksimirati ga pravokutnikom.

Promatra li se funkcija $y'(t)$ između dvije zadane točke — t_i i t_{i+1} — integral funkcije moguće je aproksimirati izračunavanjem površine pravokutnika postavljenim između točaka. Širina pravokutnika jednaka je udaljenosti između točaka



Slika 4.1: Slika vizualizira postupak aproksimacije integrala unutar intervala $[t_i, t_{i+1}]$ lijevom, odnosno desnom pravokutnom formulom

t_i i t_{i+1} , a visina pravokutnika jednaka je $y'(t_i)$ (koristi se lijeva pravokutna formula). Opisana metoda vrlo dobro odgovara motivaciji Riemannovoga integrala te ju je moguće matematički opisati izrazom

$$\int_{t_i}^{t_{i+1}} y'(t) dt \approx h y'(t_i), h = t_{i+1} - t_i. \quad (4.4)$$

Slika 4.1 prikazuje opisani postupak aproksimacije integrala pravokutnikom.

Kada se integral u izrazu 4.3 zamijeni aproksimacijom, dobiva se izraz

$$y(t_i + h) = y(t_i) + h y'(t_i)$$

Uvrsti li se 4.1 u prethodni izraz, dobiva se, uz pojednostavljeni zapis, formula

$$y_{i+1} = y_i + h f(t_i, y_i), \quad (4.5)$$

što je upravo konačan izraz za Eulerov postupak. Iz izvedenoga izraza vidljivo je da vrijednost funkcije u koraku t_{i+1} ovisi samo o prethodnome koraku. Postupke kod kojih je to slučaj nazivamo eksplicitnim postupcima.

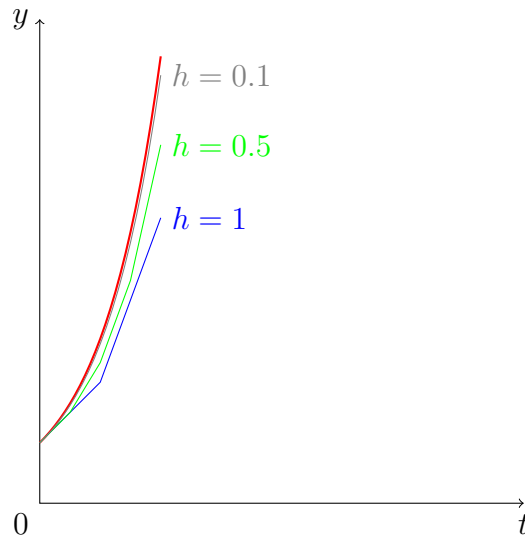
4.1.1. Preciznost

Predstavljeni izvod Eulerovog postupka tek je jedan od mogućih. Zanimljiv pristup predstavlja razvoj funkcije $y(t)$ u Taylorov red oko točke t_i jer tim načinom možemo utvrditi preciznost Eulerovog postupka.

Razvoj funkcije $y(t)$ u Taylorov red oko točke t_0 prikazujemo izrazom

$$y(t_0 + h) = y(t_0) + h y'(t_0) + \frac{1}{2} h^2 y''(t_0) + O(h^3), \quad (4.6)$$

pri čemu $O(h^3)$ predstavlja članove razvoja stupnja većeg od 2. Usporede li se izrazi 4.5 i 4.6, može se uočiti da se podudaraju u prva dva člana, a razlikuju u



Slika 4.2: Prikaz preciznosti Eulerovog postupka za jednadžbu $y' = y$, čije je egzaktno rješenje $y = e^t$, uz iznose koraka 1, 0.5 te 0.1

preostalima. Upravo zadnja dva člana razvoja određuju lokalnu pogrešku postupka — pogrešku nastalu pri svakom koraku. Ako je treća derivacija funkcije $y(t)$ ograničena, tada za $h \rightarrow 0$ član s h^2 teži k nuli brže od preostalih članova višega stupnja pa je i lokalna greška proporcionalna h^2 , što označavamo s $\mathcal{O}(h^2)$.

Budući da je prilikom izračunavanja konačne vrijednosti potrebno provesti $\frac{t_n - t_0}{h}$ koraka, što je proporcionalno s $\frac{1}{h}$, a greška u svakom koraku proporcionalna je s h^2 , globalna greška proporcionalna je s h , što označavamo s $\mathcal{O}(h)$. Dokaz ove tvrdnje moguće je istražiti u (Budin, 2002).

Na slici 4.2 moguće je vidjeti kako iznos koraka h utječe na preciznost numeričkoga rješenja. Slika vizualizira primjenu postupka nad jednadžbom $y' = y$ uz korake iznosa 1, 0.5 te 0.1. Vidljivo je da kako korak pada, tako raste i preciznost postupka. Potrebno je stoga pronaći kompromis između što manjega broja iteracija potrebnih da se dođe do rješenja te preciznosti numeričke aproksimacije.

4.1.2. Stabilnost

Numerička stabilnost Eulerovog postupka određuje se testnom jednadžbom

$$y'(t) = \lambda y, \lambda \in \mathbb{C} \quad (4.7)$$

Egzaktno rješenje testne jednadžbe jest $y(t) = e^{\lambda t}$. Uvrštavanjem testne jednadžbe u 4.5 dobiva se izraz

$$\begin{aligned} y_{i+1} &= y_i + h\lambda y_i \\ &= (1 + h\lambda)^{i+1} y_0 \end{aligned}$$

Budući da je rješenje testne jednadžbe stabilno za $\text{Real}(\lambda) < 0$, a želimo se da numeričko rješenje ponaša na jednak način, tada slijedi da mora vrijediti

$$|1 + h\lambda| < 1,$$

što je konačan uvjet stabilnosti postupka. Više detalja o izvodu i dokazu moguće je pronaći u (Budin, 2002).

Opisani postupak poznat je u literaturi još i kao eksplicitni Eulerov postupak (engl. *explicit Euler method*) te unaprijedni Eulerov postupak (engl. *forward Euler method*).

4.2. Implicitni Eulerov postupak

Implicitni Eulerov postupak vrlo je sličan (eksplicitnom) Eulerovom postupku uz jednu manju razliku: kod Eulerovog postupka integral se aproksimira lijevom pravokutnom formulom, a kod implicitnog koristi se desna pravokutna formula. Tako formula 4.4 postaje

$$\int_{t_i}^{t_{i+1}} y'(t) dt \approx hy'(t_{i+1}). \quad (4.8)$$

Uvrštavanjem izraza 4.8 u izraz 4.3 dobiva se formula implicitnog Eulerovog postupka.

$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}) \quad (4.9)$$

Vrlo je važno uočiti kako se s obje strane jednadžbe sada pojavljuje y_{i+1} , što znači da je ovaj postupak implicitan te da je rješenje moguće analitički pronaći samo u slučaju kada je $f(t_{i+1}, y_{i+1})$ linearna funkcija. U ostalim slučajevima potrebno je upotrijebiti neki numerički postupak traženja korijena realne funkcije.

4.2.1. Preciznost

Preciznost implicitnog Eulerovog postupak odgovara preciznosti eksplicitnog postupka pa je tako lokalna greška $\mathcal{O}(h^2)$, a globalna $\mathcal{O}(h)$, iz istih razloga kao što je to bio slučaj kod Eulerovog postupka.

4.2.2. Stabilnost

Prilikom ispitivanja stabilnosti također se koristi ispitna jednadžba 4.7 pa se uvrštavanjem u 4.9 dobiva izraz

$$\begin{aligned} y_{i+1} &= y_i + h\lambda y_{i+1} \\ &= \left(\frac{1}{1 - h\lambda} \right)^{i+1} y_0 \end{aligned} \quad (4.10)$$

Budući da i sada želimo da se numeričko rješenje ponaša kao egzaktno, sada mora vrijediti

$$|1 - h\lambda| > 1.$$

S obzirom na to da je korak h uvijek pozitivan, iz izraza za uvjet stabilnosti možemo uočiti da stabilnost postupka ne ovisi o koraku, nego samo o konstanti λ . Za postupke za koje ovo vrijedi kaže se da su bezuvjetno stabilni, što znači da je, iz perspektive stabilnosti, moguće odabrati proizvoljno velik korak integracije. Nažalost, ovo vrijedi samo za stabilnost, ali ne i za preciznost, tako da je svejedno nužno pažljivo kontrolirati iznos koraka h .

Za implicitni Eulerov postupak u literaturi se koristi i izraz unazadni Eulerov postupak (engl. *backward Euler method*).

4.3. Primjena na modelu tkanine

Opisani fizikalni modeli tkanine zasnivaju se na računanju različitih sila, zbrajanju sila te dobivanja akceleracije, brzine i pozicije iz temeljne jednadžbe drugog Newtonovog zakona. Budući da je ta jednadžba zapravo diferencijalna jednadžba drugoga reda, nije je moguće egzaktno riješiti na računalu. U pomoć pristižu prethodno opisani postupci rješavanja običnih diferencijalnih jednadžbi.

Prije samoga rješavanja valja uočiti da se jednadžba drugog Newtonovog zakona može zapisati pomoću dvije diferencijalne jednadžbe prvoga reda, kao što je to vidljivo u izrazu 4.11, u kojem $f(x(t), v(t))$ predstavlja izraz koji određuje konačnu akceleraciju čestice. Na diferencijalne jednadžbe prvoga reda mogu se jednostavno primijeniti opisani numerički postupci, uzimajući u obzir da je početno stanje definirano početnim stanjem čestica tkanine.

$$\frac{d}{dt} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ f(x(t), v(t)) \end{pmatrix} \quad (4.11)$$

Na model opisan jednostavnim Hookeovim zakonom moguće je primijeniti Eulerov postupak. Prije računanja nove pozicije svake čestice nužno je odrediti resultantnu silu koja djeluju na svaku česticu, a iz izračunate sile, na temelju mase čestice, određuje se ukupna akceleracija. Korištenjem Eulerovog postupka, prvo se izračunava brzina za $(n + 1)$ -vi korak, a iz brzine moguće je dobiti i novu poziciju čestice.

$$\mathbf{a}_{rez}^{n+1} = \frac{\mathbf{F}_{rez}^{n+1}}{m} \quad (4.12)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + h\mathbf{a}_{rez}^{n+1} \quad (4.13)$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + h\mathbf{v}^{n+1} \quad (4.14)$$

Izračunata pozicija koristi se prilikom iscrtavanja tkanine u novome koraku animacije, a petlja iscrtavanje–osvježavanje koristi se do kraja simulacije.

U općenitom slučaju moguće je da primjena eksplicitnih metoda, pogotovo kada se radi o sustavu masa i opruga, nije moguća. Razlog tome jest što diferencijalne jednačbe koje se rješavaju mogu biti krute (engl. *stiff equations*). To znači da korak integracije potreban da bi se dobilo stabilno rješenje mora biti puno manji nego što je to potrebno da bi rješenje bilo precizno. Više o ovome problemu moguće je pročitati u (Witkin i Baraff, 2001).

Budući da se koristi eksplicitna Eulerova metoda, potrebno je posebno obratiti pozornost na iznos koraka h , koji se u razvijenim simulacijama postavlja na recipročnu vrijednost broj sličica u sekundi. Ovakav odabir pokazao se dovoljno dobrim za postizanje uvjerljivoga ponašanja tkanine bez utjecaja na stabilnost postignutih rezultata.

5. Vizualizacija tkanine i scene

Budući da je konačan cilj izrada modela koji vjerno opisuju ponašanje tkanine, razvijene modele potrebno je i vizualizirati. Osim što se time postiže krajnja svrha kompletnoga postupka, omogućava se i verifikacija dobivenih rezultata te međusobna usporedba odabranih modela i postupaka.

Prilikom simulacije tkanine, pozicije dijela točaka tkanine ostavljaju se nepromijenjena. Na taj način moguće je vizualizirati obješenu tkaninu, što je posebno važno kada se tkanina koristi kao dio drugoga objekta ili treba biti fiksirana na nekome mjestu, u slučaju da se simuliraju objekti poput zastava ili zavjesa.

5.1. Osnovni prikaz tkanine

Bazu cjelokupnoga prikaza tkanine predstavlja rešetka sastavljena od međusobno spojenih trokuta, slično kao što je to opisano prilikom određivanja aproksimacije strukture tkanine. Uz pomoć OpenGL-ove primitive `GL_TRIANGLES` dovoljno je redom navoditi točke svih trokuta kako bi se na ekranu prikazala rešetka. Tako dobivena rešetka dovoljna je tek za provjeru početnih rezultata, stoga se vizualizacija tkanine dodatno poboljšava.

5.2. Dodavanje vizualnih efekata

Kako bi izrađeni prikaz tkanine bio što realističniji, na tkaninu se dodaje tekstura koja odgovara stvarnim materijalima za izradu tkanine. Dodatno se u generiranu scenu dodaju izvori svjetlosti kako bi se naglasili nabori na tkanini.

5.2.1. Primjena tekstura

Tekstura tkanine predstavljena je jednostavnom slikom koja prikazuje dio stvarne tkanine. Kako bi cjelokupna simulirana tkanina poprimila realističan izgled, po-

trebno je teksturu preslikati na dijelove tkanine te postupak ponavljati dok tkanina u potpunosti nije prekrivena teksturom.

Postoji širok skup tehnika koje se primjenjuju prilikom definiranja teksture objekta. Opsežniji popis uz pripadajuća objašnjenja moguće je pronaći u (Čupić i Mihajlović, 2014). Kako je tkanina sastavljena od povezanih trokuta, u ovome radu koristi se postupak uv-mapiranja.

Kod uv-mapiranja slika koja predstavlja teksturu postavi se u koordinatni sustav s osima u i v , koje se obje protežu između vrijednosti 0 i 1, uključivo. Donji lijevi kut teksture dobiva koordinatu $(0, 0)$, a gornji desni $(1, 1)$. Ove koordinate predstavljaju parametarske koordinate teksture.

Slično pridjeljivanje potrebno je provesti i za rešetku koja predstavlja tkaninu. Donjem lijevom vrhu rešetke dodjeljuje se parametarska koordinata $(0, 0)$, a gornjem desnom $(1, 1)$. Preostali vrhovi dobivaju koordinate u ovisnosti o svojoj poziciji unutar rešetke. Ako se širina i visina rešetke predstave varijablama l_x i l_y , tada će vrh s koordinatama (x, y) u lokalnom koordinatnom sustavu dobiti parametarske koordinate $(x/l_x, y/l_y)$. Navedena tvrdnja vrijedi kada se rešetka tkanine prilikom definiranja postavi tako da joj je orijentacija jednaka orijentaciji lokalnoga koordinatnog sustava, a donji lijevi vrh postavljen u ishodište.

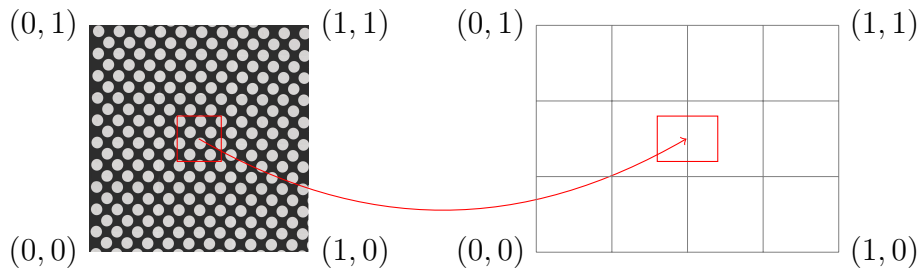
Primjenom postupka uv-mapiranja te opisanoga načina dodjeljivanja parametarskih koordinata, jednoznačno je definirano kako se tekstura preslikava na rešetku tkanine. Vrlo je važno uočiti da opisani postupak prilagođava sliku koja predstavlja tkaninu tako da u potpunosti prekriva rešetku tkanine. Ovo je valjano samo u slučaju da tekstura predstavlja kompletnu tkaninu, a ne samo jedan manji dio, odnosno uzorak tkanine. U slučaju da je tekstura tek maleni uzorak, moguće je uv-koordinate rešetke dodjeljivati tako da se dopuste i pozitivne vrijednosti izvan intervala $[0, 1]$, čime se omogućava ponavljanje teksture po tkanini.

Vizualizaciju postupka preslikavanja koordinata prikazuje slika 5.1. Na slici je dan primjer u kojem odnos širine i visine rešetke ne odgovara odnosu širine i visine teksture pa dolazi do rastezanja teksture.

5.2.2. Dodavanje svjetla

Izvor svjetlosti i sjenčanje poboljšavaju vizualni aspekt simulacije tkanine jer stvaraju osjećaj dubine. Time je jednostavnije uočiti nabore tkanine pa je i razvijena simulacija realističnija.

Za primjenu postupka sjenčanja potrebno je poznavati normale svih poligona,



Slika 5.1: Primjer načina definiranja parametarskih koordinata kod uv-mapiranja uz različit oblik teksture i rešetke te izdvojen dio preslikavanja.

odnosno normale svih vrhova poligona. Izračunate normale koriste se kako bi OpenGL mogao odrediti intenzitete svjetlosti svake točke objekta. Proračunata normala pojavljuje se OpenGL-u pozivanjem funkcije `glNormal3f`, koja prihvaća tri realna broja koji predstavljaju komponente normale. U svrhu postizanja što efektivnijih rezultata, normala se pojavljuje za svaki vrh poligona.

Prije određivanja normala u točkama poligona, potrebno je prvo odrediti normale za svaki pojedini poligon. Rešetka tkanine sastoji se od povezanih trokuta pa svaki trokut čini jednu ravninu za koje je moguće odrediti normalu. Poznato je da je ravnina jednoznačno određena trima nekolinearnim točkama P_0, P_1, P_2 koje leže na njoj (Elezović, 2012). Budući da je normala okomita na svaki vektor koji leži u ravnini, tada je sigurno i okomita na vektore koje određuje razlika radij-vektora točaka P_0 i P_1 , odnosno P_0 i P_2 . Time se dobiva da je normalu ravnine moguće izračunati vektorskim produktom navedenih vektora

$$\mathbf{n}_t = \mathbf{P}_0\mathbf{P}_1 \times \mathbf{P}_0\mathbf{P}_2,$$

pri čemu je $\mathbf{P}_0\mathbf{P}_1 = \mathbf{P}_1 - \mathbf{P}_0$, a $\mathbf{P}_0\mathbf{P}_2 = \mathbf{P}_2 - \mathbf{P}_0$. Točke P_0, P_1, P_2 u ovom konkretnom slučaju predstavljaju vrhove trokuta.

Nakon što su određene normale svih trokuta, potrebno je odrediti normale pojedinih vrhova. Budući da jedan vrh može biti dio više trokuta, normala vrha definira se kao srednja vrijednost svih normala trokuta kojima vrh pripada.

$$\mathbf{n}_v = \frac{1}{N} \sum_{i=1}^N \mathbf{n}_{t,i}$$

U prethodnom izrazu N predstavlja ukupan broj trokuta kojima vrh pripada, a $\mathbf{n}_{t,i}$ označava normalu i -toga trokuta.







Na temelju proračunatih normala vrhova te boje dobivene iz teksture, OpenGL može postupkom sjenčanja izračunati novu boju svakoga piksela koji je dio objekta.



5.3. Umetanje objekata u scenu



Osim same tkanine, u scenu je poželjno dodavati i druge objekte. Umetnuti objekti koriste se radi stvaranja dodatnih složenijih slučajeva. Objekti mogu biti pomični te se sudarati s tkaninom ili statični, ali smješteni u blizini tkanine kako bi tkanina dolazila u kontakt s objektima, što omogućava razmatranje složenih ponašanja tkanine. Posebno je zanimljivo umetanje čvrstih objekata poput stolova ili stolica kako bi se ponašanje simulirane tkanine bolje usporedilo s ponašanjem stvarne tkanine.

5.4. Upravljanje kamerom

U slučaju potpuno statičnih scena, sasvim je prihvatljivo na temelju koordinata svih objekata unaprijed odrediti volumen pogleda: poziciju kamere (očičšte), točku prema kojoj je kamera usmjerena (gledište), raspon koordinatnih osi te vektor koji određuje smjer “gore” (engl. *view-up vektor*). Kada se radi o sceni s dinamičkim objektima, inicijalne postavke volumena pogleda ne moraju biti valjanje kroz cijelu animaciju.

U tu svrhu, u razvijeni programski produkt uvodi se mogućnost upravljanja postavkama volumena pogleda mijenjanjem pozicije kamere. Omogućava se rotacija kamere oko sve tri koordinatne osi. Pritiskom na tipke  i  obavlja se rotacija kamere oko *view-up* vektora, koji je u razvijenim simulacijama uvijek definiran kao vektor $(0, 1, 0)$. Tipke  i  rotiraju kameru oko x-osi, a tipke  i  oko y-osi.

Osim rotiranja, kameru je moguće i pomicati u smjeru vektora definiranog pozicijama očičšta i gledišta korištenjem tipaka  . Prva pomiče očičšte prema gledištu, a druga ga udaljava od gledišta. Pomicanjem očičšta pomiče se i bliža ravnina volumena pogleda.

Rasponi koordinata na početku animacije definirani su unaprijed, a tijekom simulacije moguće ih je istovremeno jednoliko povećavati ili smanjivati u svim smjerovima pritiskanjem tipaka  i , što je korisno kada prvotna definicija zbog pomicanja objekata nije valjana kroz kompletno trajanje animacije.

6. Otkrivanje sudara

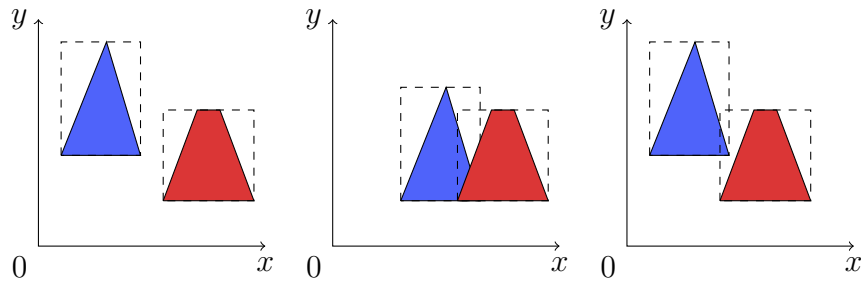
Zbog potrebe za realističnom simulacijom tkanine, potrebno je što bolje kontrolirati njezino ponašanje. Kako se tkanina sastoji od velikoga broja trokuta, nužno je pratiti njihove pozicije kako se ne bi dogodilo da tkanina prolazi sama kroz sebe. Dodatno, dodavanjem drugih objekata u scenu, potrebno je kreirati mehanizme otkrivanja i razrješavanja kolizija između tkanine i drugih objekata. U slučaju da je dodano više vanjskih objekata, javlja se i potreba za praćenjem kolizija između takvih objekata.

6.1. Široka faza otkrivanja sudara

U svrhu određivanja kolizija između različitih objekata, pa tako i između tkanine i ostalih objekata u sceni, u ovome radu koristi se binarno stablo bazirano na omeđujućem volumenu poravnatom s koordinatnim osima (engl. *axis-aligned bounding box*), takozvano AABB stablo. U daljnjem tekstu, zbog kratkoće, koriste se samo nazivi AABB (za omeđujući volumen), odnosno AABB stablo.

AABB predstavlja jednostavnu metodu u širokoj fazi detekcije kolizije koja omogućava određivanje objekata koji ne sudjeluju u koliziji. Svaki objekt predstavi se omeđujućim kvadrom (ili pravokutnikom ako se radi o 2D prostoru) čije dimenzije ovise o minimalnim i maksimalnim koordinatama objekta. Minimalne i maksimalne koordinate tada određuju dva dijagonalna vrha kvadra, što je dovoljno za određivanje kompletnoga kvadra. Prilikom ispitivanja odnosa dvaju objekata, cilj je provjeriti postoji li mogućnost da se objekti sudaraju. Utvrdi li se da je došlo do sudara, moguće je iskoristiti metode uske faze detekcije kolizije u svrhu preciznijih određivanja kolizije.

Provjera metodom AABB odvija se prema vrlo jednostavnom pravilu: objekti se sigurno ne sudaraju ako su maksimalne koordinate jednoga objekta manje od minimalnih koordinata drugoga objekta. Opisano pravilo prikazano je algoritmom 1 te slikom 6.1. U postupku provjere važno je usporediti sve komponente,



Slika 6.1: 2D primjeri određivanja kolizija metodom AABB. U prvom slučaju metoda pokazuje da ne dolazi do sudara, u drugome je sudar ispravno otkriven, a u trećem metoda daje lažno pozitivan rezultat.

što je u algoritmu pojednostavljeno korištenjem indeksa min, odnosno max. Na slici važno je uočiti treći slučaj koji pokazuje da postupak može dovesti do lažnoga pozitivnoga rezultata.

Algoritam 1 Provjeravanje sudara metodom AABB

Ulaz: A , B – objekti koji se ispituju

Izlaz: True ako se objekti sigurno ne sudaraju, inače False

if $A_{\min} < B_{\max}$ ili $B_{\min} < A_{\max}$ **then**

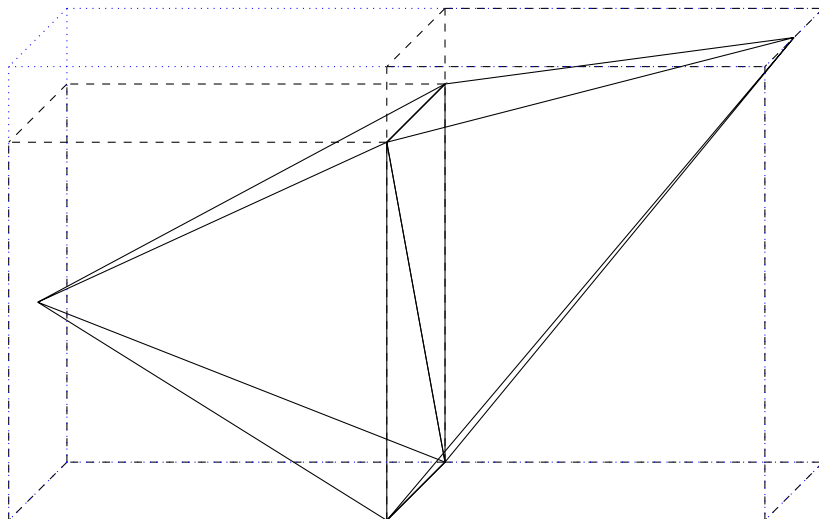
return True

else

return False

Kako je već napomenuto, AABB ima problem jer ponekad može dojaviti da postoji mogućnost sudaranja tijela, iako do kolizije ne dolazi. Problem je uzrokovan velikom pogreškom u aproksimaciji objekta. Kako bi se problem umanjio, moguće je kreirati hijerarhiju omeđujućih volumena (engl. *bounding volume hierarchy*, *BVO*). Postupak se obavlja rekurzivno počevši od kvadra koji omeđuje kompletan objekt. Početni kvadar dijeli se na nekoliko manjih koji zajedno bolje aproksimiraju tijelo. Dobiveni manji kvadri dalje se dijele dok se ne postigne dovoljno dobra preciznosti: zrnatost na razini pojedinoga poligona ili grupe poligona.

Opisanim postupkom dobiva se stablasta struktura u kojoj je čvor kvadar od kojega se krenulo, a unutrašnji čvorovi predstavljaju manje kvadrate dobivene podjelom. Konkretnih postupaka koji se temelje na ovoj ideji ima zaista mnogo, a u ovome radu odabrano je AABB stablo. Vrlo jednostavan primjer hijerarhije omeđujućih volumena temeljene na metodi AABB vidljiv je na slici 6.2 za dvije povezane piramide. Radi preglednosti, hijerarhija je izvedena do razine pojedinih



Slika 6.2: Jednostavan omeđujući volumen na primjeru spojenih piramida

piramida, a ne sve do razine pojedinih poligona.

6.1.1. AABB stablo

AABB stablo temelji se na opisanoj ideji AABB postupka jer hijerarhija omeđujućih volumena gradi se uz pomoć omeđujućih kvadara. Dobiveno stablo predstavlja balansirano binarno stablo u kojem je korijen omeđujući kvadar cijeloga objekta, unutrašnji čvorovi predstavljaju omeđujuće kvadre dijelova objekta, primjerice grupa poligona, a listovi stabla sadrže osnovne gradivne dijelove objekta. Kod tkanine, listovi sadrže pojedine trokute od kojih je tkanina načinjena.

Dodavanje elemenata

Dodavanje elemenata u stablo započinje od korijena stabla. Prilikom dodavanja potrebno je odlučiti se umeće li se novi čvor u lijevo ili desno podstablo. Nažalost, ova odluka temelji se na heuristici, a ne na strogo definiranom pravilu, tako da je u ovome radu korištena vrlo jednostavna heuristika — novi čvor umeće se u ono podstablo čiji se omeđujući volumen dodavanjem novoga čvora manje širi. Ovu provjeru moguće je odrediti uspoređivanjem volumena dvaju dobivenih podstabala. Time je cilj dobiti što kompaktnije dijelove stabla, dakle podstabla koji što bolje aproksimiraju stvarne odnose dijelova objekta. Ova heuristika odabrana je zbog jednostavnosti implementacije te dovoljno dobrih rezultata koje je na temelju nje moguće ostvariti.

Smisleno je odabrati i neku drugu heuristiku, poput dijeljenja jednoga od

intervala na nekoj osi roditelja na dva dijela te odabiranja lijevog podstabla ako je centar čvora više lijevo od centra intervala nego desno, odnosno desnog podstabla ako je centar čvora više desno od centra nego lijevo. Kod ove heuristike potrebno je dodatno i odabrati koja os će biti podijeljena na dva dijela, što se opet može odrediti korištenjem heuristike, čime se postupak dodatno komplicira.

S obzirom na to da dodavanje čvora može promijeniti omeđujući kvadar, nakon dodavanja svakog novog čvora potrebno je promijeniti vrijednosti omeđujućih kvadara u svim unutrašnjim čvorovima.

Uklanjanje elemenata

Uklanjanje elemenata iz AABB stabla predstavlja relativno jednostavnu operaciju. Cilj uklanjanja predstavljaju listovi stabla jer oni sadrže konkretne informacije o samome objektu. Budući da se gradi balansirano binarno stablo, nakon što se ukloni list stabla, njegov roditelj zamijeni se svojim drugim djetetom.

Nakon što je proveden postupak uklanjanja čvora, potrebno je ponovno izračunati omeđujuće volumene za preostale unutrašnje čvorove te za korijen stabla.

Osvježavanje stabla

Svaki pomak primitiva od kojih je objekt konstruiran može spremljene podatke učiniti netočnima jer poligoni izlaze iz granica svojih omeđujućih volumena. Kako ne bi bilo potrebno u svakome koraku animacije ponovno proračunavati omeđujuće volumene za cijelo stablo, svakom čvoru pridodaje se još jedan omeđujući volumen. On je nešto veći od stvarnoga omeđujućega volumena, a koristi se radi određivanja trenutka u kojem je potrebno osvježiti podatke u stablu. Kada jedan objekt izade izvan svojega širega omeđujućega volumena, potrebno je osvježiti njegove podatke te podatke svih čvorova prethodnika.

Dodavanjem dodatnoga omeđujućega volumena uklanja se potreba za osvježavanjem kompletnoga stabla u svakome koraku animacije, uz napomenu da je potrebno podesiti veći omeđujući volumen tako da ne bude pretjerano velik kako rezultati simulacije ne bi brzo postali netočni.

Ispitivanje kolizija

Otkrivanje sudara provodi se pretraživanjem izgrađenoga stabla. Prvo se ispituje omeđujući kvadar u korijenu stabla. Ustanovi se li da postoji mogućnost kolizije

objekata, ispituju se djeca korijena. Rezultat ispitivanja djece jest rezultat dobiven testiranjem podstabla za koji je određeno da je uzrok sudara. Uoči li se mogući sudar za neko podstablo, nastavlja se dalje s grananjem sve dok se ne dođe do listova stabla ili dok se u nekom trenutku ne ustanovi da je na prethodnim razinama došlo do pogreške te da objekti nisu u koliziji.

Iz perspektive traženja dijelova objekta koji sudjeluju u koliziji za samo jednu česticu tkanine te objekt s n primitiva, moguće je postići logaritamsku vremensku složenost — $\mathcal{O}(\log n)$, što je značajno ubrzanje u odnosu na $\mathcal{O}(n)$ kada se radi ispitivanje svih dijelova objekta zasebno. Pretraživanje za kompletnu tkaninu s m čestica tada ima složenost $\mathcal{O}(m \log n)$.

6.2. Uska faza otkrivanja sudara

Nakon što su pronađeni dijelovi tkanine i objekti koji potencijalno sudjeluju u koliziji, potrebno je provesti precizno utvrđivanje sudara. U tu svrhu koristi se uska faza ispitivanja kolizija, koja dio otkrivenih sudara uklanja jer neki otkriveni sudari predstavljaju “lažnu uzbunu”, a preostale potvrđuje te otkriva dodatne potrebne informacije za razrješavanje sudara.

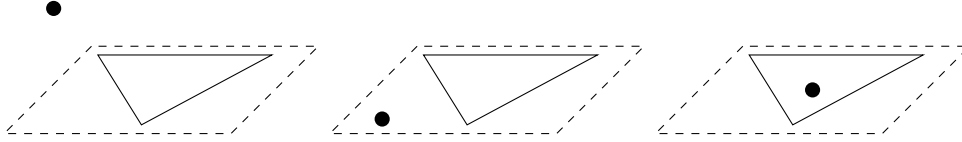
Uska faza ispitivanja kolizija korištena u ovome radu bazira se na traženju sjecišta točke i poligona. Nakon što su osvježene pozicije svih čestica tkanine te svih objekata, za svaku česticu poznaje se njezina trenutno iscrtana pozicija te nova pozicija koja će biti korištena prilikom iscrtavanja u novoj slici. Trenutna i nova pozicija određuju pravac pa vrijedi izraz

$$\mathbf{P}_{\text{novi}} = \mathbf{P} + t\mathbf{c}, \quad (6.1)$$

pri čemu \mathbf{P}_{novi} označava novu poziciju čestice, \mathbf{P} staru poziciju čestice, \mathbf{c} smjer kretanja po pravcu da bi se od trenutne točke došlo do nove, a t označava vremenski korak potreban za ostvarivanje željenoga pomaka. Budući da su \mathbf{P} i \mathbf{P}_{novi} poznate, a poznat je i t jer označava korak integracije, tada je vrlo jednostavno izračunati \mathbf{c} pomoću izraza

$$\mathbf{c} = \frac{\mathbf{P}_{\text{novi}} - \mathbf{P}}{t}. \quad (6.2)$$

Ako je došlo do kolizije točke i poligona, tada postoji točka Q na ravnini određenoj poligonom. Kako se ta točka nalazi između točaka \mathbf{P}_{novi} i \mathbf{P} , tada i za nju vrijedi jednadžba 6.1. Kako bi se dobila točna pozicija sjecišta Q , potrebno je odrediti



Slika 6.3: Mogući slučajevi prilikom uske faze otkrivanja sudara: a) čestica se ne nalazi ni u ravnini pa tako ni unutar trokuta, b) čestica se nalazi na ravnini, ali ne unutar trokuta, c) čestica se nalazi unutar trokuta

parametar t_Q koji opisuje vremenski trenutak u kojem se kolizija događa. Budući da točka Q leži u ravnini određenoj poligonom, tada ona zadovoljava jednadžbu ravnine $Ax + By + Cz + D = 0$, pri čemu su A , B i C komponente normale ravnine pa ih je moguće odrediti vektorskim produkt dvaju vektora definiranih trima vrhovima poligona. Nakon određivanja parametara ravnine, parametarska jednadžba točke Q uvrštava se u jednadžbu ravnine, čime se dobiva izraz

$$\begin{aligned}
 \mathbf{n} \cdot \mathbf{Q} + D &= 0 \\
 \mathbf{n} \cdot (\mathbf{P} + t_Q \mathbf{c}) + D &= 0 \\
 \mathbf{n} \cdot \mathbf{P} + t_Q \mathbf{n} \cdot \mathbf{c} + D &= 0 \\
 t_Q \mathbf{n} \cdot \mathbf{c} &= -\mathbf{n} \cdot \mathbf{P} - D \\
 t_Q &= \frac{-(\mathbf{n} \cdot \mathbf{P} + D)}{\mathbf{n} \cdot \mathbf{c}}. \tag{6.3}
 \end{aligned}$$

Sudar čestice i poligona događa se kada nazivnik izraza 6.3 poprimi vrijednost različitu od nule. S obzirom na to da je izraz zapravo dobiven računanjem sjecišta pravca i ravnine, potrebno je provjeriti nalazi li se točka u poligonu. S obzirom na to da su i tkanina i svi objekti u ovome radu predstavljeni trokutima, prvo se uvrštavanjem izračunate vrijednosti t_Q u parametarsku jednadžbu sjecišta dobiva točka Q , a zatim se računaju baricentrične koordinate točke Q za trokut s kojim se ispitivao sudar. Postupak nalaženja baricentričnih koordinata opisan je u (Čupić i Mihajlović, 2014). Ako su sve baricentrične koordinate u intervalu $(0, 1)$, čestica se nalazi unutar trokuta. Slika 6.3 ilustrira slučajeve koji se mogu pojaviti. Radi preciznosti, valja napomenuti da je moguće da se čestica nalazi i na samome rubu trokuta. Prilikom razrješavanja sudara uzima se u obzir i taj slučaj, a prepoznaje se po tome što je jedna baricentrična koordinata jednaka jedan.

Opisanim postupkom moguće je otkrivati sudare između tkanine i drugih objekata, ali i unutar same tkanine. Nakon što su određeni svi objekti koji sudjeluju u kolizijama, razrješavaju se svi pronađeni sudari.

7. Razrješavanje sudara

Detekcija sudara predstavlja jedan od najvažnijih dijelova implementacije realističnih ponašanja ne samo tkanine, nego bilo kojega dijela stvarnoga svijeta. No, sama detekcija sudara nije dovoljna jer nakon što su otkriveni objekti koji sudjeluju u koliziji, potrebno je odrediti što će se s objektima dogoditi. Nikako nije prihvatljivo dopustiti da objekti prolaze kroz tkaninu ili tkanina kroz objekte. Osim toga, tijekom animacije mogu se dogoditi pomaci koji bi dio tkanine pokušali gurnuti kroz samu tkaninu, što također nije željeni efekt.

Detekciju i razrješavanje sudara potrebno je odvojiti jer temelje se na dva različita principa. Detekcija kolizija spada u probleme računalne geometrije jer jedina zadaća otkrivanja sudara jest otkriti nalazi li se točka ili neki geometrijski lik ili objekt unutar drugoga lika ili objekta. Razrješavanje sudara predstavlja, pak, fizikalni problem: sudjelovanje u sudaru mijenja pozicije i akceleracije objekata sukladno fizikalnim zakonima, primjerice sukladno zakonu očuvanja gibanja. Nadalje, nije svejedno sudjeluje li u sudaru tkanina načinjena od svile, koja, zbog niskoga koeficijenta trenja, vrlo lako može skliznuti s objekta na koji pada, ili pak komad tkanine načinjen od vune, koji se zbog visokoga koeficijenta trenja vune lakše zadržava na površini na koju pada.

Prilikom razrješavanja kolizija treba voditi računa o tome radi li se o koliziji između tkanine i nepomičnoga objekta od kojeg se tkanina samo odbija ili pak o sudaru tkanine i pomičnoga objekta koji se, zbog svoje značajno veće mase, nastavlja gibati kao i do trenutka kolizije, ali i mijenja ponašanje tkanine.

Kako je naglasak ovoga rada na simulaciji ponašanja same tkanine, razmatraju se sudari tkanine s pomičnim ili nepomičnim objektom te sudari unutar same tkanine. U scenu se nikada uz tkaninu ne dodaje više od jednoga pomičnoga objekta pa se postupci razrješavanja sudara dvaju ili više objekata od kojih niti jedan nije tkanina niti ne razmatraju u ovome radu, iako se opisani koncepti mogu proširiti i na taj slučaj. U svim postupcima opisuje se sudar između čestice tkanine te trokuta jer se time pokrivaju oba tipa sudara koji su od značajnoga

interesa. Dodatno, odabrane su vrlo jednostavne, ali dovoljno dobre metode razrješavanja kolizija opisane u (Kieran et al., 2005), iako bi kod još složenijih simulacija tkanine prikladnije bilo koristiti tehnike poput onih opisanih u (Sifakis et al., 2008), (Brochu et al., 2012).

7.1. Sudari tkanine s nepomičnim objektima

Kada se tkanina sudara s nepomičnim objektom, tada objekt jednostavno ima ulogu “prepreke” od koje se tkanina odbija. Odbijanje od objekta moguće je postići na više načina, a među jednostavnije spadaju promjena pozicije čestice tako da čestica izlazi iz objekta te promjena brzine čestice tako da se čestica sama u sljedećem trenutku odmake od pozicije u kojoj dolazi do sudara. S obzirom na to da proizvoljni nagli pomaci čestica tkanine mogu dovesti do nestabilnosti u numeričkom postupku određivanja stanja tkanine, primjenjuje se korekcija brzine čestice te se nova pozicija ponovno računa na temelju promijenjene brzine.

Čestica tkanine P kreće se brzinom \mathbf{v} prema trokutu čija je normala \mathbf{n} . Kada je sudar detektiran, brzina \mathbf{v} korigira se mijenjanjem komponente brzine koja je paralelna normali \mathbf{n} trokuta. Označi li se ta komponenta brzine s \mathbf{v}_n , tada je ulaznu brzinu čestice moguće razložiti na dvije komponente:

$$\mathbf{v} = \mathbf{v}_n + \mathbf{v}_t. \quad (7.1)$$

Komponenta \mathbf{v}_t ostavlja se nepromijenjenom, a komponenti \mathbf{v}_n mijenja se orijentacija te iznos množenjem s koeficijentom $-k_r$, koji označava restitucijski koeficijent. On određuje u kojoj mjeri sudar elastičan, odnosno neelastičan. Postavi li se vrijednost koeficijenta na 1, tada je sudar savršeno elastičan, a u svim ostalim slučajevima postaje neelastičan. Nova brzina određena je dakle izrazom

$$\mathbf{v}_{novo} = \mathbf{v}_t - k_r \mathbf{v}_n. \quad (7.2)$$

Restitucijski faktor uglavnom se ne postavlja na najveću moguću vrijednost jer ni u stvarnome svijetu kod sudara tkanine i drugih objekata ne radi se o savršeno elastičnim sudarima, nego tkanina u određenoj mjeri prijanja uz površinu objekta pri čemu dolazi do određenoga gubitka energije. Posebnu pažnju valja posvetiti vrijednostima kada se modelira padanje tkanine na čvrste objekte poput stola ili poda jer tada se očekuje da će tkanina ostati na površini objekta ili lagano skliznuti s nje, ali ne i da će se odbiti od površine.

7.2. Sudari tkanine s pomičnim objektima

U ovome slučaju razrješavanja sudara uzima se u obzir jednostavna pretpostavka: masa objekta značajno je veća od mase tkanine pa se objekt nastavlja kretati kao što se kretao i prije sudara (količina gibanja objekta ostaje očuvana), ali zato sudar mijenja ponašanje tkanine. Kako bi ova pretpostavka iznjedrila što vjerodostojnije ponašanje tkanine nakon sudara, koristi se zakon očuvanja količine gibanja.

Tijekom postupka detekcije kolizije otkrivena je točka Q koja predstavlja sjecište između trokuta i zrake određene starom i novom pozicijom čestice tkanine. S obzirom na to da se presjecište nalazi na trokutu koji se pomiče, tada se pomiče i ta točke pa samim time ima određenu brzinu pa i količinu gibanja. Sudar između čestice tkanine i trokuta događa se upravo na pronađenome sjecištu pa je potrebno odrediti u kojoj mjeri sudar utječe na količinu gibanja čestice tkanine. Za trokut su unaprijed poznate brzine u njegovim vrhovima pa se brzina u točki sjecišta može izračunati uz pomoć linearne interpolacije utemeljene na baricentričnim koordinata. Ako su poznate brzine $\mathbf{v}_A, \mathbf{v}_B, \mathbf{v}_C$ trokuta ABC, a baricentrične koordinate točke Q neka su označene varijablama t_1, t_2, t_3 , tada brzina u točki Q iznosi

$$\mathbf{v}_Q = t_1 \mathbf{v}_A + t_2 \mathbf{v}_B + t_3 \mathbf{v}_C. \quad (7.3)$$

Nakon što je određena brzina u točki Q , koristi se izraz sličan izrazu 7.2, samo što se u novi iznos brzine uvodi i brzina točke Q modificirana restitucijskim koeficijentom k_r koji određuje koliko energije se raspršuje prilikom sudara, odnosno je li nastupio elastičan ili neelastičan sudar, uostalom kao je to bio i slučaj u jednadžbi 7.2. Razlika je tek u činjenici da se komponenta \mathbf{v}_n zamjenjuje komponentom $\mathbf{v}_{Q,n}$ pa konačan izraz za brzinu tako postaje

$$\mathbf{v}_{novo} = \mathbf{v} - \mathbf{v}_n + \mathbf{v}_{Q,n}. \quad (7.4)$$

U izvedenom izrazu moguće je još brzinu $\mathbf{v}_{Q,n}$ pomnožiti restitucijskim koeficijentom kako bi se modelirao stupanj (ne)elastičnosti sudara.

Važno je imati na umu da su opisani postupci rezolucije kolizija vrlo jednostavni pa je moguće da nakon što se razriješi dio kolizija u jednome dijelu tkanije, kao rezultat se pojave kolizije u drugome dijelu tkanine, što ni u jednom trenutku nije bilo moguće detektirati pa samim time niti obraditi. U složenijim animacijama to bi svakako trebalo uzeti u obzir.

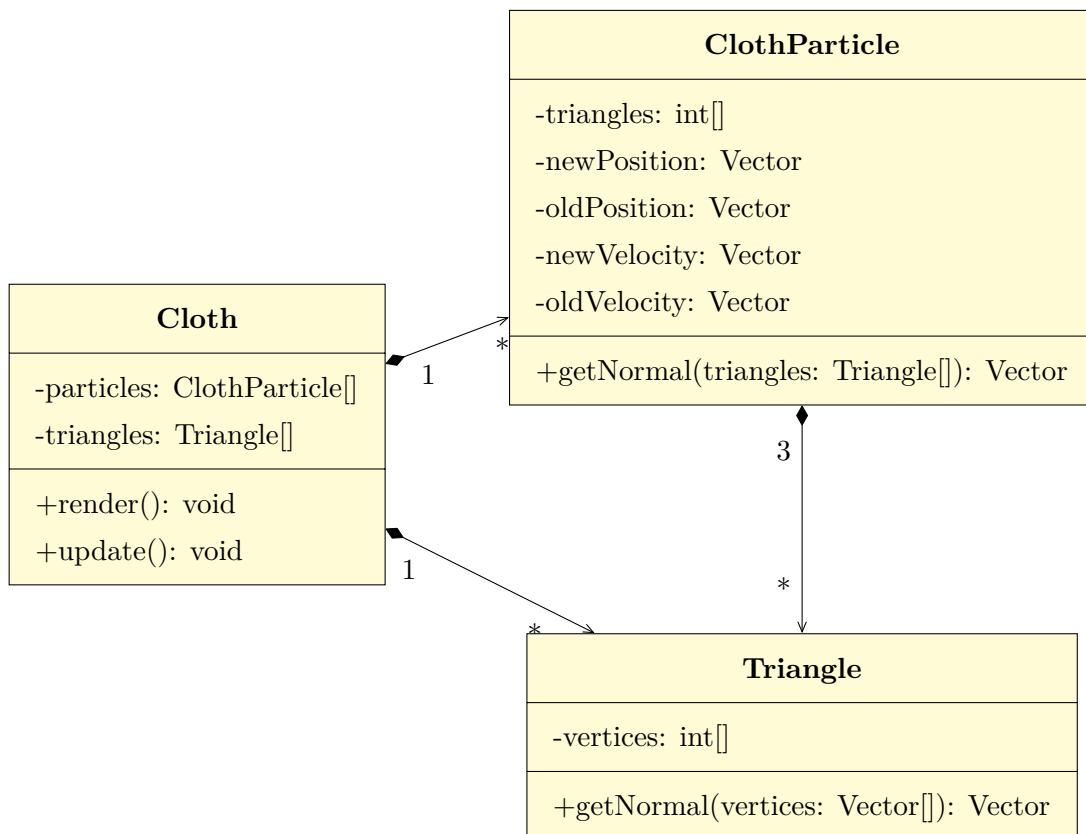
8. Programska implementacija

Programska podrška razvijena za simulaciju opisanih modela tkanine ostvarena je programskim jezikom Java i uz pomoć biblioteke JOGL, koja pruža programsko sučelje za korištenje OpenGL-a unutar programa pisanih u jeziku Java.

8.1. Učitavanje podataka

Vizualizacija modela tkanine započinje učitavanjem objekata koji su dio scene. Izgrađena scena može sadržavati statičke ili dinamičke objekte. Tipovi objekata označavaju se u kodu korištenjem sučelja `FixedObject`, odnosno `MoveableObject`. Tkanina je modelirana razredom `Cloth` i ona predstavlja dinamički objekt jer pozicije čestica tkanine mijenjaju se tijekom simulacije. Korišteni statički objekti tijekom izrade animacije često su objekti poput poda, kada se ispituje ponašanje tkanine tijekom pada na ravnu površinu, ili stola ako je cilj ispitati kako se tkanina prilagođava obliku čvrstoga tijela.

Neovisno o tipu, svi objekti spremljeni su unutar objekta koji je instanca razreda `Scene`. Taj objekt koristi se za spremanje svih važnih podataka o sceni, pa samim time i svih objekata koji sudjeluju u animaciji. Kako bi krajnji programski produkt bio što robusniji, definicije svih objekata spremljene su u Wavefront OBJ datoteci (`Wav`). Korištenje vanjske datoteke omogućava veću robusnost programskoga produkta jer glavni program može raditi s bilo kojim objektom, umjesto da se definicije objekata striktno definiraju unutar samoga programa. Radi dodatnoga povećanja robusnosti programskoga produkta, podaci o sceni (popis objekata scene, inicijalne postavke kamere te opisi svjetala) spremljeni su u posebnoj datoteci s ekstenzijom `.scene`, a definicije unutar datoteke napisane su jezikom YAML (`YAM`). Datoteke slične namjene, ali puno ekspresivnije sintakse, koriste se primjerice u razvojnom okruženju Unity.



Slika 8.1: Dijagram razreda koji prikazuje osnovnu strukturu razreda potrebnih za vizualizaciju tkanine. Navedeni popis metoda i svojstava nije potpun.

8.2. Vizualizacija rezultata

Budući da tkanina predstavlja sustav masa i opruga, a mase su predstavljene česticama postavljenima u rešetku definiranu trokutima, objekt razreda `Cloth` sadrži dva polja; jedno polje sadrži objekte stvorene na temelju razreda `Triangle`, tako da svaki objekt sadrži indekse čestica koje određuju trokut; drugo polje sadrži sve čestice tkanine. Čestica tkanine predstavljena je razredom `ClothParticle`. Kako bi tijekom simulacije bilo moguće sjenčati tkaninu korištenjem normala vrhova trokuta, nije dovoljno samo unutar trokuta imati referencu na čestice, nego i svaka čestica treba imati reference na sve trokute čiji je dio. Slika 8.1 predstavlja izdvojena svojstva i metode opisanih razreda te njihove međusobne odnose. Valja dodatno uočiti da iako na dijagramu razreda postoji veza između razreda `ClothParticle` i `Triangle`, ona je navedena samo ilustrativno jer oba razreda sadrže cjelobrojne indekse, a ne stvarne reference.

U poglavlju 5.4 navedeno je da je tijekom simulacije moguće upravljati kamerom. Kako bi se omogućila ta funkcionalnost, podaci o kameri spremljeni su

unutar razreda `Camera`. Kamera je definirana svojom pozicijom (očiste) te točkom u koju gleda (gledište). Kako bi se omogućila željena funkcionalnost, razred nudi metode `move` i `rotate`. Metoda `move` prima jedan parametar koji određuje smjer pomaka kamere: `CAMERA_IN` pomiče očiste prema gledištu, a `CAMERA_OUT` udaljava očiste od gledišta. Metoda `rotate` prima dva parametra: jedan parametar određuje os rotacije (`AXIS_X`, `AXIS_Y`, `AXIS_Z`), a drugi stupanj rotacije zadan u stupnjevima.

8.3. Izračunavanje novih stanja objekata

Pozicije svih pokretnih objekata potrebno je osvježiti prije iscrtavanja novoga stanja animacije. Ažuriranje čestica tkanine prepušteno je razredu `MassSpringSystem`, odnosno dvama izvedenim razredima — `SpringBending` i `AngleBending`. Prvi razred koristi se za ažuriranje pozicija čestica u modelu kod kojeg se i za reakciju na savijanje tkanine koriste opruge, a drugi za model kod kojeg je savijanje modelirano kutom između dvaju trokuta. Oba izvedena razreda kroz konstruktor primaju instancu razreda `Cloth`, čije se čestice mijenjaju u svakom novom koraku. Nakon što je dobiven objekt koji predstavlja tkaninu, kreira se polje veza koje predstavljaju opruge unutar tkanine. Svaka veza predstavljena je razredom `Connection`, a u njemu su pohranjene čestice koje određuju oprugu te parametri opruge (dužina l i konstanta elastičnosti k).

Kako bi postupak mogao poznavati željeni iznos vremenskoga perioda integracije h , taj podatak također se definira kao parametar konstruktora. Definicije sile koje djeluju na čestice tkanine predaju se pozivanjem metoda `addParticleForce`, odnosno `addSpringForce`, ovisno o tome djeluje li sila na česticu (primjerice sila teža) ili je rezultat pomicanja čestica koje definiraju oprugu (primjerice elastična sila opruge).

Vrlo je važno da se pozicije čestice ne mijenjaju odmah nakon što su izračunate sve njezine sile, nego tek nakon što su izračunate sve sile za sve čestice. U protivnome nakon što je osvježena pozicija čestice i , računanje sile za česticu j vršilo bi se između dva različita vremenska koraka, što bi dovelo do netočnih rezultata.

U svakome novom koraku osvježavanja tkanine na svaku česticu tkaninu primjenjuju se pojedinačne sile te sile koje djeluju između čestica tkanine te se na kraju osvježavaju pozicije svih čestica. Opisani postupak moguće je predočiti isječkom programskog koda 8.1.

Programski isječak 8.1: Osvježavanje pozicija čestica

```
1 for(Connection connection: connections) {
2     ClothParticle p1 = connection.particle1;
3     ClothParticle p2 = connection.particle2;
4     Spring spring = connection.spring;
5
6     p1.addForce(force.apply(p1, p2, spring));
7 }
8
9 for(ClothParticle particle: clothParticles) {
10     for(ParticleForce force: particleForces) {
11         particle.addForce(force.apply(particle));
12     }
13 }
14
15 for(ClothParticle particle: clothParticles) {
16     // osvjezi poziciju cestice na temelju odabranog
17     // numerickog postupka
18 }
```

8.4. Otkrivanje sudara

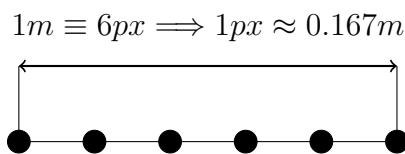
Široka faza otkrivanja sudara implementirana je AABB stablom, koje je predstavljeno razredom `AABBTree`. Korijski čvor i svi unutrašnji čvorovi enkapsulirani su razredom `Node`, a svi listovi stabla razredom `LeafNode`. Razlika između tipova čvorova načinjena je kako bi bilo moguće u listove dodatno pohraniti i prošireni omeđujući volumen za potrebe osvježavanja stabla tijekom animacije, kao što je opisano u potpoglavlju 6.1.1. Svaki sudar pronađen tijekom široke faze sprema se uz pomoć tablice raspršenoga adresiranja (`HashMap`), pri čemu je ključ jednak indeksu čestice tkanine koja sudjeluje u koliziji, a vrijednost je trokut s kojim je čestica u koliziji. Nikakve druge posebne strukture podataka ili algoritmi nisu potrebni za ostvarivanje opisanih postupaka u uskoj fazi te tijekom razrješavanja.

9. Utjecaji parametara na simulaciju

Prilikom opisivanja modela tkanine javljaju se različiti koeficijenti čije je vrijednosti potrebno odrediti. U fizikalnome modelu nužno je, primjerice, poznavati duljine i konstante opruga te iznose faktora prigušenja. Tijekom određivanja odgovora na kolizije koristi se restitucijski faktor koji utječe na elastičnost sudara pa je i to vrijednost koju je potrebno unaprijed definirati. Na kraju, kako bi izgled tkanine bio što realističniji, modeliraju se utjecaji materijala na sjenčanje tkanine: nije svejedno je li komad tkanine načinjen od svile ili pak vune.

Na prvi pogled čini se da je adekvatno definiranje parametara modela tkanine sasvim dovoljno da bi dobivena simulacija bila kvalitetna. Ta tvrdnja točna je ako je ispunjen vrlo važan preduvjet: sve jednadžbe moraju imati vrijednosti čije su jedinice iz SI sustava. Budući da su pozicije objekata definirane u pikselima, nužno je odrediti odnos između jednoga piksela i jednoga metra. Jedna mogućnost jest odrediti taj odnos dinamički i to na temelju dimenzija tkanine. Primjerice, pretpostavi se da je stvarna tkanina koja se oponaša uvijek široka i dugačka točno jedan metar. Dodatno, pretpostavi se i da je broj horizontalnih i vertikalnih čestica također jednak. Jedino što se mijenja jest broj čestica tkanine pa se odnos piksela i metra određuje tako da se jedan metar podijeli udaljenošću između dvije krajnje horizontalne ili vertikalne čestice, što je moguće vidjeti na slici 9.1.

Jednostavniji pristup navedenoga problema, a koji se i koristi u ovome radu, jest jednostavno zanemariti činjenicu da izračunate udaljenosti nisu izražene u



Slika 9.1: Primjer određivanja odnosa između piksela i metra

metrima. To je moguće jer, iz perspektive mjere udaljenosti, sustav scene predstavlja sustav ekvivalentan stvarnome svijetu; sva ostala svojstva poput mase, vremena te ostalih jedinica zadržavaju svoju originalnu interpretaciju. Dakle, jednostavno se sve jedinice koje uključuju metre zamijene jedinicama koje uključuju piksele. Tako, primjerice, m/s postaje px/s , a m/s^2 postaje px/s^2 . Osim toga, ovaj pristup podosta pojednostavnjuje računanja jer nije potrebno uzimati u obzir složena pretvaranja jedinica, što je posebno zahtjevno kada su dimenzije tkanine različite po visini i širini.

Dodatno, prilikom simulacije ponekad je korisno povećati broj čestica tkanine kako bi se lakše uočili specifični nabori u ispitnim situacijama. No, kako to ne bi značilo i veliko povećanje samoga objekta unutar scene, prilikom generiranja tkanine visina i širina tkanine fiksiraju se na točno određenu veličinu, a onda povećanje broja čestica samo smanjuje njihov međusobni razmak.

9.1. Utjecaj na oblik i ponašanje tkanine

Budući da se prilikom izračunavanja akceleracije koja djeluje na čestice koristi Newtonov zakon, evidentno je da svaka čestica mora imati definiranu masu. Kako bi to bilo moguće izvesti, unaprijed se definira stvarna dimenzija komada tkanine koji se simulira te se na temelju gustoće materijala jednoliko raspodijeli masa po svim česticama tkanine. Primjerice, za materijal “White Dots on Blk” u radu (Wang et al., 2011) definirana je površinska gustoća $\rho = 0.128kg/m^2$. Simulirala se komad tkanine površine jednoga kvadratnog metra uz stotinu čestica, tada se dobiva da bi masa svake čestice, odluči li se primijeniti jednolika distribucija mase, trebala biti $m_i = \frac{0.128}{100} = 0.00128$ kg. Tako definirana masa čestice može biti veliki problem prilikom simulacije. Naime, s obzirom na to da je u svakome koraku potrebno izračunati akceleraciju i -te čestice formulom $\mathbf{a}_i = \frac{\mathbf{F}_i}{m_i}$, vidljivo je da će izračunata akceleracija biti izrazito velika čak i za maleni iznos sile jer nazivnik je jako malena vrijednost, pogotovo što je broj čestica tkanine veći. Budući da je u stvarnim primjenama broj čestica značajno velik, opisani postupak nije moguće izravno primijeniti. Kako bi se izbjegao taj problem, masa svake čestice postavlja se na vrijednost koja je do jedan ili dva reda veličine veća od proračunate vrijednosti ili se, pak, za svaki materijal posebno definira masa čestice određena na temelju ispitivanja i evaluacije dobivenih rezultata.

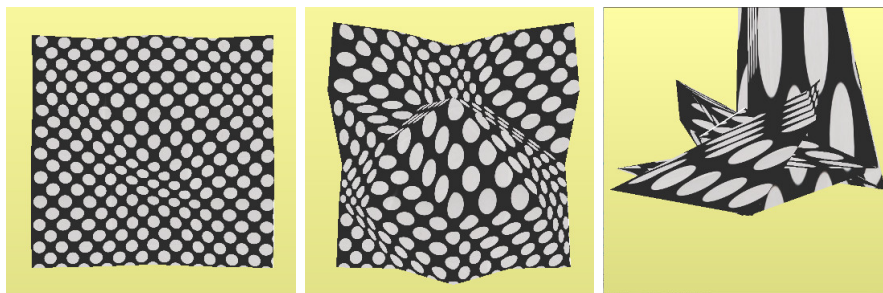
9.1.1. Svojstva opruga

Svaka opruga tkanine definirana je svojom konstantom te duljinom. Određivanje duljine opruge nije problem jer svaka opruga nalazi se između neke dvije čestice pa tako udaljenosti čestica u početnoj konfiguraciji tkanine određuju i duljine opruga. U nekim radovima početne duljine opruga postavljaju se na nulu, tako da sile opruga djeluju u svakome trenutku. No, to nije slučaj u ovome radu.

Konstante opruga predstavljaju nešto veći problem. Nažalost, većina radova iz područja računalne grafike koji se bave modeliranjem tkanine ne navode parametre korištene prilikom izrade simulacija. Ova činjenica možda nije toliko ni neobična uzme li se u obzir da je osnovni cilj postići rezultate koji se samo doimaju stvarnima pa konačne vrijednosti parametara ovisе i o subjektivnoj procjeni kvalitete postignutih rezultata. Ipak, na sreću postoje radovi poput (Wu et al., 2003) i (Wang et al., 2011) koji se bave i postupcima određivanja parametara tkanine pa je vrijednosti navedene u njima moguće koristiti kao dobar početak. Nažalost, navedene vrijednosti uglavnom je moguće primijeniti tek ako se preslika i korišteni model tkanine.

U ovome radu konstante opruge određivane su empirijski: evaluacijom vizualnih rezultata te proučavanjem stabilnosti tkanine za različite vrijednosti parametara. Na temelju opaženih rezultata zaključuje se da konstante opruge trebaju biti vrlo male jer inače dolazi do nestabilnosti numeričkih postupaka, što se u simulaciji može manifestirati “eksplozijom tkanine”. Ovaj neželjeni efekt prikazuje slika 9.2. Naime, zbog nepreciznosti i nestabilnosti postupka te neadekvatnoga perioda integracije dolazi do sve snažnijih titranja čestica oko njihovih ravnotežnih položaja, sve dok se čestice na posljetku ne “razlete”. To je upravo primjer manifestacije krutih jednadžbi, kao što je opisano u potpoglavu 4.3, a uzrokovane su suviše čvrstim oprugama. Na prikazanim slikama vidljivo je kako se prvo počinju javljati nabori unutar tkanine uzrokovani titranjem čestica oko ravnotežnih položaja, a zatim dolazi i do potpunoga raspada tkanine.

Adekvatan odabir konstanta opruga (zasebno za svaki tip opruge) značajno utječe i na izgled dobivenih rezultata. Konstanta koja se primjenjuje prilikom računanja reakcijske sile na savijanja utječe na izraženost nabora u tkanini: što je konstanta opruge veća, to su i nabori izraženiji kada se na tkaninu primijene vanjske sile. Postavljanje niskih vrijednosti za vlačne opruge učinit će tkaninu slabo otpornom na rastezanja pa se može dogoditi da sila teža te ostale vanjske sile počnu tkaninu rastezati kao da se radi o komadu gume. Vrijednosti odabrane za



Slika 9.2: Eksplozija tkanine uzrokovana nestabilnošću numeričkoga postupka za odabrane parametre

smične opruge određuju, naravno, koliko se tkanina opire smičnim naprezanjima, što je vizualno vrlo dobro moguće vidjeti blizu fiksiranih čestica. Naime, kada je konstanta smičnih opruga visoka, obješena počinje se i uvijati između fiksiranih točaka umjesto da se samo rasteže.

Prilikom izračunavanja elastičnih sila te novih pozicija čestica potrebno je uzeti u obzir i stvarna svojstva tkanina. Naime, lako je opaziti da se većina tkanina vrlo snažno opire vlačnim i smičnim naprezanjima, a za savijanja vrijedi upravo suprotno. Ovo ponašanje moguće je modelirati snažnijim oprugama, ali problem je što tada sile postaju značajno velike pa se kod fiksiranih čestica javljaju znatno duža produljena, što u konačnom rezultatu izgleda kao “topljenje” tkanine pa tkanina izgleda kao da je načinjena od gumenoga materijala.

Kako bi se riješio navedeni problem, uvodi se parametar koji određuju maksimalnu dopuštenu elongaciju opruga, primjerice 10%. Dogodi li se da se dvije čestice udalje jedna od druge za više od 10% inicijalne duljine opruge, njihova udaljenost se smanjuje kako bi se zadovoljio postavljeni kriterij. Valja napomenuti da popravljavanje udaljenosti između dviju čestica može dovesti do toga da čestice čije su pozicije već korigirane ponovno postanu previše udaljene ili pak suviše bliske. No, praktični rezultati pokazali su da ovaj postupak rješava više problema nego što ih unosi, tako da je primijenjen u implementaciji.

9.1.2. Otpor zraka

Modeliranje sile otpora zraka važno je u svrhu što vjernijega kretanja tkanine kroz scenu. Oblikovanje ove sile moguće je načiniti na temelju izraza 3.3, pri čemu se parametrom b opisuje viskoznost sredstava u kojem se tkanina nalazi. Niske vrijednosti parametar prigušenja koriste se za definiranje otpora zraka, što je uglavnom i slučaj koji se želi modelirati. Uz visoke vrijednosti faktor

prigušenja dobio bi se dojam da se tkanina kreće kroz vrlo gusti fluid, što najčešće nije ponašanje koje se želi postići, stoga je odabiranje adekvatnoga parametara najbolje učiniti na temelju eksperimentalnih rezultata.

9.2. Postupak provjeravanja parametara i implementacije

Inicijalno pokretanje razvijene simulacije, osim u slučaju velike sreće s odabirom parametara, može dati vrlo loše rezultate. Uzrok toj činjenici jest velik broj parametara koje je potrebno odrediti, a ne treba niti zanemariti mogućnost pogrešne implementacije nekoga dijela sustava. Ispitivanje i rješavanje potonjega problema ne spada u potpunosti u okvir ovoga rada, ali s obzirom da jedan problem može vrlo lako zasjeniti drugi, navodi se nekoliko jednostavnih pravila na temelju kojih je moguće provjeriti je li problem nastao zbog neadekvatnih parametara ili grešaka u programskome kodu.

Prvi korak jest ukloniti utjecaj svih sila, osim elastičnih sila koje djeluju zbog prisutnosti opruga. Neovisno o odabiru parametara opruga, dobivena animacija trebala bi biti sasvim statična. Naime, s obzirom na to da se elastične sile javljaju tek kada dođe do elongacije opruge, a nikakva vanjska sila koja bi to mogla prouzročiti nije pristuna, sve elastične sile trebaju biti jednake nuli te ne smije doći do promjena pozicija čestica. Ovime se provjerava odgovaraju li duljine opruga inicijalnim udaljenostima između čestica.

U drugome koraku uvodi se sila teža. Kako je ta sila globalna za cijelu tkaninu, primjena sile na svaku česticu pojedinačno mora dovesti do trivijalnoga ponašanja: sve čestice tkanine trebaju se jednoliko kretati u smjeru vektora \mathbf{v}_{dolje} ¹. Činjenica da se sve čestice gibaju jednoliko znači da ni u ovome slučaju ne smije doći do pojave elastičnih sila jer udaljenosti između čestica ostaju očuvane.

Nakon što je ustanovljeno da sila teža uzrokuje očekivano ponašanje, fiksiraju se neke pozicije čestica tkanine. To znači da neovisno o silama koje djeluju u sceni te unutar same tkanine, pozicije fiksiranih čestica ne smiju se promijeniti. Dakle, sile na njima ne djeluju. Takve čestice vrlo su važne za određivanje konstanti opruga. Budući da su neke čestice statične, pomaci preostalih čestica dovode do elongacija opruga, što znači da se pojavljuju elastične sile. Sila teža nastoji čestice

¹Vektor \mathbf{v}_{dolje} označava smjer koji u simulaciji odgovara izrazu “dolje”, a najčešće je to vektor $(0, -1, 0)$.

odvući prema dolje, a elastične sile opiru se tome. Dobivena simulacija trebala bi prikazati to ponašanje izmjenjivanjem smjera kretanja tkanine: prvo pomicanjem prema dolje zbog utjecaja sile teže, a zatim vraćanjem prema gore zbog djelovanja elastičnih sila. Dobiveno ponašanje treba se neprestano ponavljati jer na tkaninu ne djeluje niti jedna sila koja bi opisano ponašanje zaustavila. U ovome trenutku mogu se pojaviti simptomi problema s numeričkom metodom: tkanina se nakon nekog vremena može zaustaviti jer greške numeričkih postupaka mogu djelovati kao sile prigušenja, ali moguće je i da tkanina počne sve snažnije titrati te da dođe do već opisane “eksplozije tkanine”. U idealnome slučaju, ne javlja se niti jedan od ova dva problema, nego tkanina djeluje kao neprigušeni titrajni sustav.

Kako bi se tkanina, u slučaju da ne djeluju dodatne sile, u nekome trenutku ipak zaustavila, dodaju se sile prigušenja. U ovome koraku rezultate je jednostavno provjeriti jer treba vidjeti dolazi li do usporavanja i zaustavljanja kretanja ili ne. Zbog netočne implementacije moguće je ponovno vidjeti efekt eksplozije, što, iz iskustva prikupljenoga tijekom implementacije ovoga rada, može značiti da je krivo određen smjer djelovanja sile prigušenja.

Točnost utjecaja preostalih sila vrlo je jednostavno vizualno provjeriti jer osnovne sile koje djeluju na tkaninu trebala bi biti točno implementirane te definirane adekvatnim parametrima.

Osim kod izračunavanja novih stanja čestica, parametri imaju utjecaj i na korektnost rezolucije kolizija. Naime, tamo je definiran restitucijski faktor koji određuje tip sudara: elastični ili neelastični. Jednostavnost vizualne provjere rezultata znači da nije potrebno provoditi detaljnije postupke testiranja jer može se vrlo lako uočiti odbija li se čestica od objekta, prijanja li uz njega ili pak nastavlja prolaziti kroz objekt. Važno je tek obratiti pozornost na činjenicu da postupci razrješavanja sudara koji direktno mijenjaju pozicije čestica mogu također dovesti do eksplozija tkanine jer takvi “ručni” pomaci mogu uzrokovati previše nagle pomake, što uzrokuje velike sile pa samim time i nestabilnosti i nepreciznosti. Budući da su u ovome radu opisani postupci razrješavanja sudara koji se temelje na korekciji brzine čestice, ne bi se trebali pojavljivati tako ekstremni problemi.

10. Potencijalna poboljšanja

Širok skup razvijenih metoda za simuliranje ponašanja tkanine znači da je u ovome radu obuhvaćen samo manji podskup poznatih postupaka. Osim toga, s obzirom na to da su raspoloživo vrijeme i znanje nažalost uvijek ograničeni, nije moguće uvijek odabrati najbolje metode jer one bi zbog svoje složenosti zahtjevale previše vremena. Stoga se ovdje predlažu neka potencijalna poboljšanja za nastavak rada na problemu realistične simulacije tkanine zasnovane na fizikalnim modelima.

U konačnici, cilj je razviti programsku podršku koja bi korisnicima na vrlo jednostavan način omogućila učitavanje postojećih modela tkanine ili crtanje modela unutar samoga programa. Dodatno, za svaki model tkanine na raspolaganju bi bio širok skup parametara koji pobliže određuju ponašanje i izgled modela, a u pozadini program bi se sam pobrinuo da se koriste fizikalni modeli i numeričke metode koje mogu pružiti maksimalno zadovoljavajuće rezultate.

10.1. Fizikalni modeli i numeričke metode

Tijekom godina razvijen je širok skup fizikalnih modela koji nastoje što vjernije opisati postupke simulacije tkanine. Budući da niti jedan od modela nije moguće proglasiti apsolutno dominantnim u odnosu na ostale, bilo bi dobro implementirati još pokoji model kako bi se usporedili s onima opisanima u ovome radu radi te kako bi se pronašle prednosti i nedostatak pojedinih modela.

Problem numeričkoga rješavanja običnih diferencijalnih jednadžbi vrlo je dobro istražen pa tako postoji mnogo numeričkih metoda koje je moguće primijeniti na probleme koji se sreću prilikom simulacije tkanine. Budući da je u ovome radu u konačnici odabran vrlo jednostavan eksplicitni postupak, tada uvođenje nekog od implicitnih postupaka može značiti dodatna poboljšanja u ostvarenim rezultatima. Iako je programska izvedba u tom slučaju ponešto kompleksnija, moguća su potencijalno ubrzanja i značajno veća preciznost.

Korištenje složenijih postupaka, poput implicitnih Runge-Kutta višega reda uz metode pronalaženja rješenja jednadži poput Newton-Raphsonovog postupka, ima potencijala dati vrlo kvalitetne rezultate, kao što je i opisano u radovima koji se detaljnije bave upravo postupcima izračunavanja stanja tkanine, primjerice (Baraff i Witkin, 1998) i (Liu et al., 2013). Nadalje, primjena složenijih implicitnih metoda omogućava definiranje adaptivnih vremenskih koraka, što može dodatno ubrzati izračunavanje novih stanja jer kreće se s pretpostavkom da je moguće primijeniti velik vremenski korak, koji se korigira — povećava ili smanjuje — ovisno o izračunatim vrijednostima.

10.2. Otkrivanje i razrješavanje sudara

Efikasno otkrivanje i razrješavanje sudara predstavlja jedan od najvažnijih djelova većine svih simulacija, neovisno o složenosti objekata u sceni. Samim time, niti simulacija tkanine nije iznimka. Nažalost, ovi postupci jedni su i od najkompleksnijih u cijelome procesu simulacije pa je vrlo važno odabrati adekvatne metode.

Kako bi krajnji rezultat bio što kvalitetniji, nije loše jednostavne postupke detekcije i rezolucije kolizija opisanih u ovome radu zamijeniti složenijima. Iako bi to vjerojatno usporilo trajanje svih izračuna, konačan rezultat mogao bi uključivati veći broj različitih tkanina te velik broj drugih objekata bez bojazni od pojavljivanja fizikalno neprihvatljivih rezultata.

Među zanimljive postupke svakako spadaju oni temeljeni na poljima udaljenosti i funkcijama skupa razina, kao što je primjerice opisano u (Breen et al., 2003).

10.3. Sjenčanje tkanine

Oblikovanje realističnoga ponašanja tkanine moguće je postići fizikalnim modelima, ali potpuno realističan izgled tkanine moguće je postići tek uz vrlo kvalitetno sjenčanje tkanine. Osim primjene tekstura i definirana normala vrhova rešetke, za kvalitetne rezultate potrebno je definirati i svojstva materijala prilikom dodavanja svjetala u scenu. Kako bi se naglasila struktura materijala poput primjerice vune, valjalo bi uzeti u obzir i primjenu mapa normala kako bi se dobio osjećaj za strukturu korištenoga materijala.

Budući da je moguće kreirati vlastite programe za sjenčanje vrhova, istraživanje takvih mogućnosti moglo bi dovesti veće slobode u određivanju izgleda tkanine pa samim time i do potencijalno zanimljivijih rezultata.

U konačnom programskom produktu pronađena svojstva materijala, teksture, mape normala i slično mogli bi se iskoristiti za stvaranje baze putem koje bi se korisnicima omogućilo vrlo jednostavno stvaranje realističnoga modela tkanine.

10.4. Napredniji ispitni slučajevi

Jednostavnim ispitivanjem ponašanja razvijene simulacije tkanine dobiva se dojam da je postupak sam sebi svrha, što nije točno. Razvoj postupaka animacije tkanine javlja se zbog realnih potreba u, primjerice, filmskoj industriji te industriji video igara. Samim time, stvara se potreba za osmišljavanjem složenijih ispitnih slučajeva ponašanja tkanine. Bilo bi vrlo zanimljivo animirati virtualnoga lika odjevenoga u odjeću koja se temelji na razvijenom modelu tkanine jer to je često ionako očekivani konačan rezultat. Dodatno, umetanjem složenih objekata koji su izvana načinjeni od komada tkanine, poput lutaka i plišanih igračaka, uvela bi se vrlo kvalitetna metoda evaluacije ostvarenih rezultata. To bi samim time i poslužilo kao dodatna motivacija u traženju što boljih postupaka simulacije tkanine.

11. Zaključak

Animacije objekata uključenih u scenu ne mogu odavati dojam vjerne kopije stvarnoga svijeta ako neka od komponenata objekta nije vjerno preslikana u virtualni svijet. Samim time uspješno ostvarenje realističnoga prikaza tkanine predstavlja važan zadatak animatora i programera zaduženih za razvoj specijalnih efekata u filmovima, razvoj računalnih igara te ostalim primjenama računalne grafike čiji je tkanina često esencijalni dio.

Iako je struktura tkanine vrlo složena, a ponašanja unutar tkanine vrlo kompleksno te nedovoljno istraženo, pokazuje se da pojednostavljeni fizikalni modeli mogu vrlo dobro poslužiti u stvaranju kvalitetnoga preslikavanja stvarnoga ponašanja u virtualni svijet. No, konačna kvaliteta modela tkanine u primijenjenoj fizikalnoj aproksimaciji ovisi i o dobro odabranim parametrima simuliranih materijala jer nije svejedno koristi li se komad platna načinjen od svile, vune ili traperera.

Vrlo zahvalnim poboljšanjima ponašanja tkanine pokazalo se uvođenje dodatnih ograničenja i veza tkanine s drugim objektima ili tkanine sa samom sobom. Tako je tkaninu moguće zalijepiti za neki dio objekta ili spojiti dijelove tkanine kako bi se ponašanje što bolje podudaralo s idejama animatora. Takve metode vrlo su važne kada je tkanina dio drugoga objekta, primjerice virtualnoga lika ili nekakvoga oblika navlake.

Kako tkanina uglavnom nije jedini objekt koji se nastoji simulirati u danome trenutku, robusni postupci otkrivanja i razrješavanja sudara nužni su radi postizanja kvalitetnih rezultata. Prodiranje tkanine kroz samu sebe ili druge objekte narušilo bi integritet i vjerodostojnost izrađene animacije, čime bi sav prethodno uloženi trud naprosto pao u vodu.

Kao konačan dodir u implementaciji realističnoga modela tkanine dolazi postupak primjene teksture te sjenčanja tkanine. Time se dodatno osnažuje dojam realističnosti prikaza, stoga je i na taj dio simulacije potrebno potrošiti značajnu količinu vremena.

LITERATURA

- Wavefront obj file format summary. <http://www.fileformat.info/format/wavefrontobj/egff.htm>. Pristupljeno: 14.6.2014.,
- Yaml ain't markup language. <http://www.yaml.org/spec/1.2/spec.html>. Pristupljeno: 14.6.2014.
- David Baraff i Andrew Witkin. Large steps in cloth simulation. U *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, stranice 43–54, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi: 10.1145/280814.280821. URL <http://doi.acm.org/10.1145/280814.280821>.
- D. Breen, M. Lin (editors, R. Bridson, S. Marino, i R. Fedkiw. Simulation of clothing with folds and wrinkles, 2003.
- Tyson Brochu, Essex Edwards, i Robert Bridson. Efficient geometrically exact continuous collision detection. *ACM Trans. Graph.*, 31(4):96:1–96:7, Srpanj 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185592. URL <http://doi.acm.org/10.1145/2185520.2185592>.
- L. Budin. Analiza i projektiranje računalom. Skripta s predavanjima, 2002.
- N. Elezović. *Linearna algebra*. Element, Zagreb, Hrvatska, 2012.
- N. Elezović, D. Žubrinić, i A. Aglič Aljinović. *Linearna algebra*. Element, Zagreb, Hrvatska, 2012.
- Jonathan M. Kaldor, Doug L. James, i Steve Marschner. Simulating knitted cloth at the yarn level. *ACM Trans. Graph.*, 27(3):65:1–65:9, Kolovoz 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360664. URL <http://doi.acm.org/10.1145/1360612.1360664>.

- Jonathan M. Kaldor, Doug L. James, i Steve Marschner. Efficient yarn-based cloth with adaptive contact linearization. U *In ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'10 conference, ACM, 2010.*
- E. Kieran, G. Harisson, i L. Openshaw. Cloth simulation. Magistarski rad, NCCA Bournemouth University, 2005.
- Tosiyasu L. Kunii i Hironobu Gotoda. Singularity theoretical modeling and animation of garment wrinkle formation processes. *Vis. Comput.*, 6(6):326–336, Studeni 1990. ISSN 0178-2789. doi: 10.1007/BF01901019. URL <http://dx.doi.org/10.1007/BF01901019>.
- Tiantian Liu, Adam W. Bargteil, James F. O'Brien, i Ladislav Kavan. Fast simulation of mass-spring systems. *ACM Transactions on Graphics*, 32(6):209:1–7, Studeni 2013. URL <http://graphics.berkeley.edu/papers/Liu-FSM-2013-11/>. Proceedings of ACM SIGGRAPH Asia 2013, Hong Kong.
- Hing N. Ng i Richard L. Grimsdale. Computer graphics techniques for modeling cloth. *IEEE Comput. Graph. Appl.*, 16(5):28–41, Rujan 1996. ISSN 0272-1716. doi: 10.1109/38.536273. URL <http://dx.doi.org/10.1109/38.536273>.
- Isaac J. Rudomin. *Simulating Cloth Using a Mixed Geometric-physical Method*. Doktorska disertacija, Philadelphia, PA, USA, 1990. AAI9112615.
- Eftychios Sifakis, Sebastian Marino, i Joseph Teran. Globally coupled collision handling using volume preserving impulses. U *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '08*, stranice 147–153, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. ISBN 978-3-905674-10-1. URL <http://dl.acm.org/citation.cfm?id=1632592.1632614>.
- Huamin Wang, Ravi Ramamoorthi, i James F. O'Brien. Data-driven elastic models for cloth: Modeling and measurement. *ACM Transactions on Graphics*, 30(4):71:1–11, Srpanj 2011. URL <http://graphics.berkeley.edu/papers/Wang-DDE-2011-08/>. Proceedings of ACM SIGGRAPH 2011, Vancouver, BC Canada.
- Jerry Weil. The synthesis of cloth objects. *SIGGRAPH Comput. Graph.*, 20(4):49–54, Kolovoz 1986. ISSN 0097-8930. doi: 10.1145/15886.15891. URL <http://doi.acm.org/10.1145/15886.15891>.

- Andrew Witkin i David Baraff. Physically based modeling: Principles and practice. Online Siggraph 2001 Course notes, 2001.
- Z Wu, CK Au, i Matthew Yuen. Mechanical properties of fabric materials for draping simulation. *International Journal of Clothing Science and Technology*, 15(1):56–68, 2003.
- Cem Yuksel, Jonathan M. Kaldor, Doug L. James, i Steve Marschner. Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Trans. Graph.*, 31(4):37:1–37:12, Srpanj 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185533. URL <http://doi.acm.org/10.1145/2185520.2185533>.
- M. Čupić i Ž. Mihajlović. *Interaktivna računalna grafika kroz primjere u OpenGL-u*. 2014.

Fizikalno temeljen model tkanine

Sažetak

U radu se opisuje postupak simulacije tkanine zasnovan na fizikalnim zakonima, prvenstveno Hookeovom zakonu, te sustavu masa i opruga. Struktura tkanine modelira se diskretnom rešetkom, a ponašanje tkanine uvjetovano je međudjelovanjima unutar rešetke te primjeni vanjskih sila.

Budući da su fizikalni zakoni predstavljeni diferencijalnim jednadžbama, proučavaju se jednostavne numeričke metode za rješavanje postavljenih jednadžbi te se ispituje utjecat odabira parametara na kvalitetu krajnjih rezultata.

U razvijenu simulaciju dodaju se i drugi objekti radi kreiranja složenijih ispitnih slučajeva, stoga se opisuju i postupci otkrivanja i razrješavanja sudara.

Ključne riječi: fizikalna simulacija tkanine, sustav masa i čestica, Hookeov zakon, diferencijalne jednadžbe, numeričke metode, otkrivanje sudara, razrješavanje sudara, AABB stablo

Physically Based Cloth Model

Abstract

This thesis describes physical models for simulating cloth based on Hooke's law and mass-spring systems. Cloth's structure is simplified using a discrete grid of particles, which is used for modeling both macroscopic and microscopic behavior of cloth.

Since physical systems are described using differential equations and because computers are not able to find exact results of those equations, some numerical methods are introduced. The importance of choosing good enough parameters is discussed throughout the text.

To enrich developed simulations, rigid bodies are added into the scene, which calls for methods for collision detection and collision response.

Keywords: physically-based cloth simulation, mass-spring system, Hooke's law, differential equations, numerical methods, collision detection, collision response, AABB tree