

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 495

**POSTUPCI SIMULACIJE VLAKNASTIH  
OBJEKATA**

Hrvoje Pađen

Zagreb, lipanj 2009.



## Sadržaj

1. Uvod .....	1
2. Pregled poznatijih metoda modeliranja vlaknastih objekata .....	2
2.1 Modeliranje poligonima i teksturama .....	2
2.2 Model čestica povezanih oprugama .....	4
2.3 Model slojeva tekstura .....	5
2.4 Metoda super zavojnica .....	7
3. Teorijska razrada modela čestica povezanih oprugama .....	10
3.1 Osnovni pojmovi .....	10
3.2 Prikaz sila koje djeluju u modelu .....	11
3.3 Utjecaj sila na brzinu i položaj čvorova .....	13
4. Programsko rješenje modela čestica povezanih oprugama .....	15
4.1 Upotrijebljene tehnologije .....	15
4.2 Programska implementacija čestice .....	16
4.3 Programska implementacija opruge .....	17
4.4 Simuliranje čestica povezanih oprugama .....	18
4.5. Krajnji rezultat .....	19
5. Zaključak .....	20
6. Popis slika .....	22
7. Literatura .....	23
8. Sažetak .....	24

# 1. Uvod

Modeliranje vlaknastih objekata u svijetu grafike je oduvijek predstavljalo problem. Elemente poput kose, krzna ili trave je uvijek bilo teško prikazati u 3D okruženju zbog raznih svojstava koja se zbog realnosti moraju zadovoljavati, a koje je teško računalno izvesti. Neke od glavnih prepreka efektnom ostvarenju su :

- Vlastita dinamika svakog pojedinog vlaknastog objekta
- Međusobna kolizija dvaju i više vlaknastih objekata
- Prirodne karakteristike vlaknastih objekata (npr. viskoznost, čvrstoća) na koje mogu utjecati i vanjski efekti (npr. kosa se drukčije ponaša kad je mokra)
- Kolizija vlaknastih objekata s drugim nevlaknastim objektima
- Kompleksni efekti svjetla i sjene

S obzirom da se realističan prikaz vlaknastih objekata zahtijeva u sve većem broju područja kao što su npr. filmska industrija (slika 1.) i industrija videoigara (slika 2.), razvile su se razne tehnike realističnog prikazivanja vlaknastih objekata (ponajprije kose i krzna). Ovaj rad će ukratko obraditi neke od tih tehnika. Iznesti će se principi rada pojedinih modela te njihovi nedostaci.



Slika 1. Primjena simulacije krzna u filmskoj industriji.



Slika 2. Primjena simulacije kose u industriji videoigara.

## 2. Pregled poznatijih metoda modeliranja vlaknastih objekata

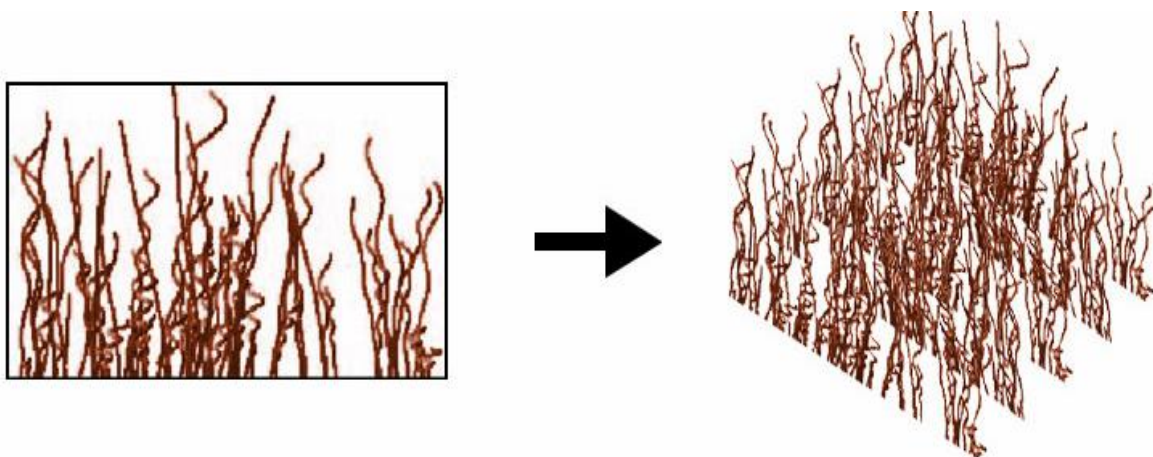
### 2.1 Modeliranje poligonima i teksturama

Modeliranje kose 3D modelima s malim brojem poligona je najjednostavnija tehnika modeliranja koja je najpopularnija u izradi modela za 3D video igre. Rabe se jednostavni modeli s malim brojem poligona na koje se potom „lijepo“ teksture kose izrađene u programima za izradu slike ili fotografije prave kose obrađene u programima za obradu slike. Najčešće se radi više manjih modela koji se potom postavljaju na model glave na način da budu što bliže jedan drugom kako bi se dobilo na volumenu kose. Ovakav način izrade kose se prakticirao u skoro svim video igrama do prije par godina. Danas je isto dosta zastupljen zbog toga što daje zadovoljavajuće rezultate uz relativno malo trošenje računalnih resursa (slika 3.). Njegove glavne mane su nerealističnost oblika uzrokovana malim brojem poligona (npr. neki pramenovi nemaju konstantnu zakrivljenost već se mijenjaju pod kutom) te nedovoljna nezavisnost kose (npr. ne može se kretati vlas po vlas, već se kreće poligon po poligon što dovodi do nerealističnih situacija). Unatoč ovim ograničenjima, ovaj model je zastupljen na mjestima gdje su brz izračun i brza animacija važnija od realističnosti prikaza.



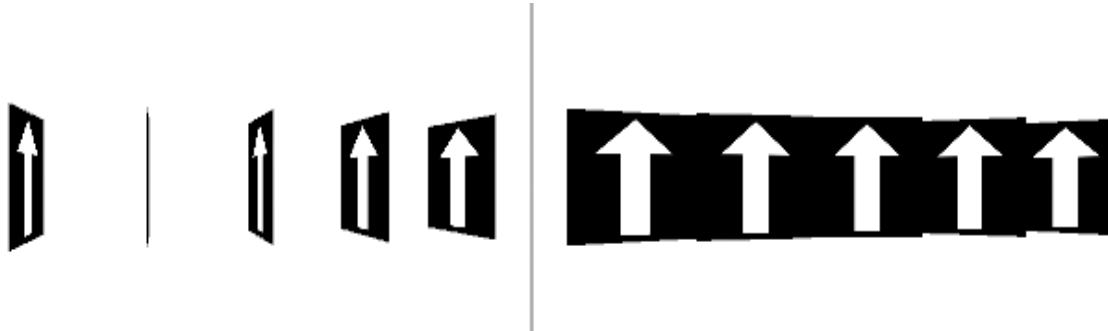
Slika 3. Modeliranje poligonima.

Modeliranje trakama poligona (eng. polygon strips) je jedna od najstarijih tehnika prikaza vlaknastih objekata poput kose ili krzna. Ovom tehnikom se rade teksture koje u profilu izgledaju kao pojedinačne vlasi kose. Potom se prikazuju na vertikalno posloženim poligonima. Poligoni se slažu u više slojeva kako bi se dobio dojam gustoće kose (slika 4.).



Slika 4. Prikaz modeliranja trakama poligona

Glavni problem kod ove tehnike je što se kod različitih kutova gledanja mogu vidjeti praznine između pojedinih slojeva (kao što je prikazano na slici). Ovaj problem se može riješiti ukošavanjem pojedinih tekstura ili razbijanjem teksture u mnogo manjih koje se onda mogu postaviti na različite pozicije i time simulirati gustoću kose. Problem ovog rješenja je što se relativno zadovoljavajući rezultati postižu jedino uz veliko trošenje računalnih resursa. Druge rješenje je tehnika *Billboardinga*. Njome se pojedini slojevi mogu okretati prema kameri, čineći tako privid pokrivenosti iz bilo kojeg kuta kad se gleda postrance. Problem i dalje ostaje pogled odozgora jer u *Billboard* tehnici se teksture mogu okretati samo oko osi koja je normala u odnosu na objekt na kojem su smješteni (slika 5.).

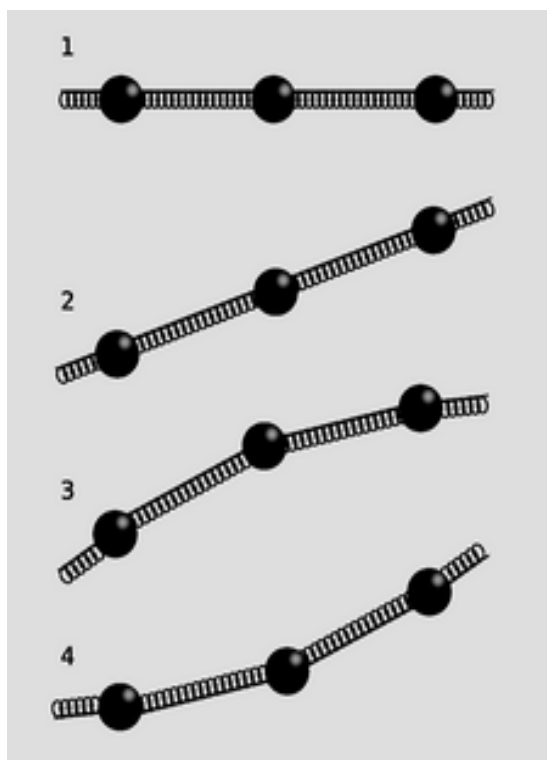


Slika 5. Obično postavljanje 2D slike (lijevo) i Billboard tehnika (desno).

## **2.2 Model čestica povezanih oprugama**

U modeliranju raznih objekata se kao vrlo zahvalan koncept pokazao model čestica povezanih oprugama. Svoju primjenu je našao uglavnom u simulacijama mekih objekata poput tkanine te u simulacijama vlaknastih objekata.

Sam koncept je relativno jednostavan, temelji se na nizu od  $n$  čestica koje su povezane oprugama (slika 6.). Za svaku česticu se računa sila koja utječe na nju te se prema tome računa i konačna pozicija čestice. U računanju sile se koristi Newtonova mehanika. Nakon izračuna pozicija svih čestica, pristupa se iscrtavanju.



Slika 6. Koncept čestica povezanih oprugama.

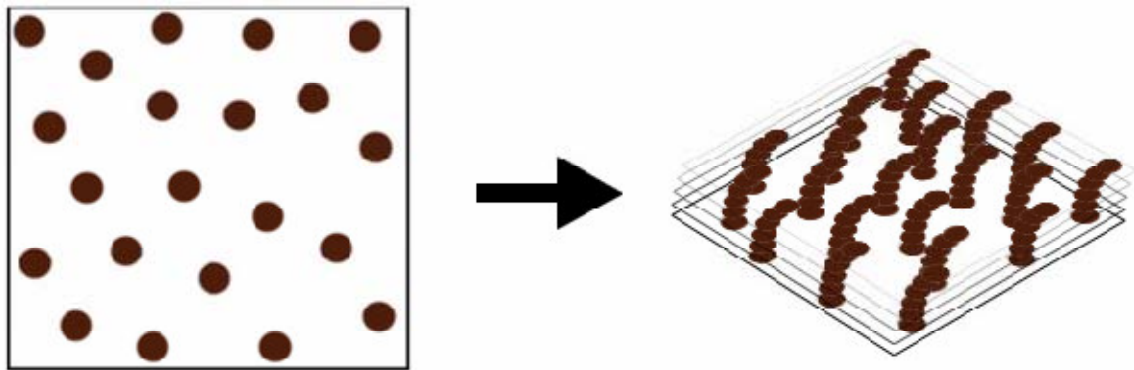
Same opruge nemaju masu, međutim utječu na kretanje svojom elastičnošću. Sile koje utječu na mase mogu se podijeliti na unutarnje sile i vanjske sile. Unutarnje sile nastaju zbog elastičnosti opruga dok vanjske mogu nastati zbog različitih posljedica kao što su gravitacija ili trenje. Ovaj model će biti temeljitije razrađen kasnije u radu.

### **2.3 Model slojeva tekstura**

Model slojeva tekstura razrađen je krajem 1998. godine od strane A. Meyera i F. Neyreta [4]. Malo kasnije je J. Lengyel upotrijebio njihov postupak kako bi simulirao krzno kod životinja.

Glavna ideja ovog modela se temelji na prikazu više slojeva istih tekstura. Te teksture su generalno prozirne osim na dijelovima na kojima želimo prikazati vlasi. Postavljanjem tekstura jedne iznad druge tj. stvaranjem niza vodoravno postavljenih tekstura dobiva se dojam volumena (slika 7.).

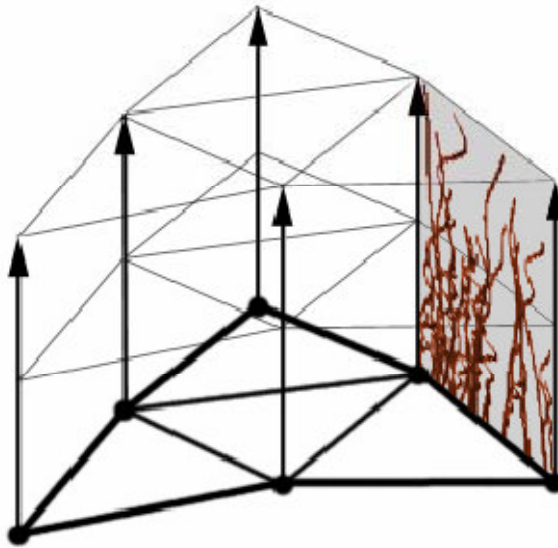




Slika 7. Model slojeva tekstura.

Model slojeva tekstura se pokazao dobrim kod simuliranja vlaknastih objekata kad se ne dolazi preblizu samim objektima. Naime, između dvije susjedne teksture u nizu uvijek ostane praznog prostora (kao što se vidi na slici) koji je vidljiv samo u slučaju kad se očiste približi na vrlo malu udaljenost od objekta ili kad dolazi do izražene devijacije objekta uzrokovane vanjskim efektom koji zadamo (npr. gravitacijskom silom). Ovo je moguće donekle riješiti povećavanjem broja tekstura, ali u slučaju prevelikog broja tekstura dolazi do znatnog pada performansi usred velikog trošenja računalnih resursa. Također, problem nastaje i kad se ovakav model gleda iz profila. Kako je svaka od tekstura beskonačno tanka, gledanjem iz profila se vidi mnogo malih „crtica“ na mjestima gdje teksture nisu prozirne.

Potonji problem je riješio J.Lengyel. On je predložio da se od prvog (najdonjeg) sloja pa do zadnjeg (najgornjeg) sloja u nizu postavi tekstura kose. Na taj način bi se pri pogledu iz profila dobro relativno dobro vidjela kosa. Ovo rješenje je prikazano na slici 8.

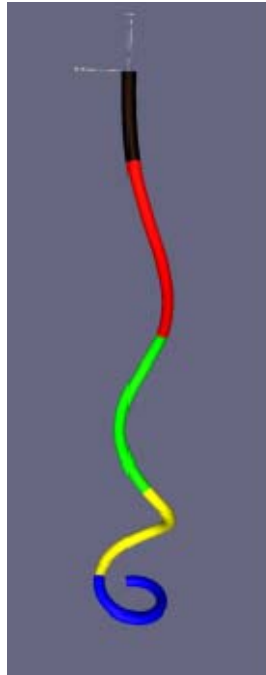


Slika 8. Rješenje problema modela slojeva tekstura.

## **2.4 Metoda super zavojnica**

Metoda super zavojnica je relativno nova metoda simulacije vlaknastih objekata, posebice kose. Razvijena je u suradnji s kozmetičkom tvrtkom L'oreal te je do danas tehnika koja najrealističnije prikazuje kosu. Važan naglasak je bio na laboratorijskom ispitivanju svojstava kose kod različitih rasnih skupina. Na taj način prikupljeni su podaci koji su omogućili bolje razumijevanje dinamičkih i optičkih karakteristika kose. Također, metoda super zavojnica je prva za koju se može reći da realno simulira kovrčavu kosu, što s tehnikama koje su ranije spomenute to ili uopće nije moguće ili su rezultati nezadovoljavajući.

Tehnika super zavojnica se temelji na prikazu kose na temelju Kirchoffove i Cosseratove teorije o motkama. Motka u ovom slučaju predstavlja dugačak elastičan materijal kojemu je duljina puno veća od veličine poprečnog presjeka. Vlas kose se predstavlja s više povezanih motki (slika 9.)



Slika 9. Vlas sastavljena od 5 motki

Svaka od ovih motki ima svoj stupanj slobode koji predstavlja zakrivljenost materijala te po kojemu se može odrediti kako će motka izgledati. Na temelju toga se određuju generalizirane koordinate pomoću kojih se u bilo kojem trenutku može odrediti oblik i pozicija motke. Osim generaliziranih koordinata, određuju se još i razni drugi parametri, poput mase vlasi, njene krutosti, prirodne kovrčavosti, unutarnjeg trenja među vlasima itd. Upotrebljavajući Lagrangeove zakone dinamike te uz navedene konstante, izrađen je prilično realističan model kose (slika 10.). Povećanjem broja motki potrebnih za jednu vlas se može dobiti na većoj realističnosti, makar pri većem broju motki promjene u dinamici kose postaju nezamjetljive.



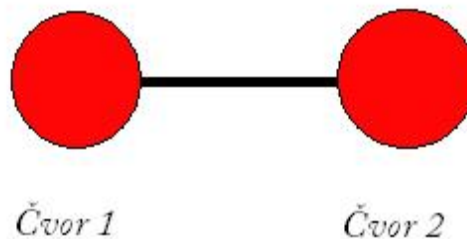
Slika 10. Usporedba realnog ponašanja kose i ponašanja u modelu super zavojnica.

Problem ovog modela je njegova velika potrošnja programskih resursa. Tako kod simulacije pune glave kose vrijeme za iscrtavanje varira između 0.3 do 3 sekunde po okviru (frameu). Također, povećavanjem broja motki vremenska složenost algoritma raste s kvadratom tog broja. Ovo zapravo i ne predstavlja problem jer se zadovoljavajući rezultati mogu postići već i za manji broj motki. Uzevši u obzir konstantno povećanje snage računala, ovaj model se nameće kao najbolje rješenje za realističnu simulaciju svih vrsta kose.

### 3. Teorijska razrada modela čestica povezanih oprugama

#### 3.1 Osnovni pojmovi

Model čestica povezanih oprugom u najjednostavnijem obliku može se prikazati kao dva čvora između kojih se nalazi elastična opruga koja ih spaja (slika 11.).



Slika 11. Pojednostavljeni prikaz modela čestica povezanih oprugama.

Svaki od čvorova ima 4 vrijednosti koje ga definiraju:

- Masu
- Poziciju u trodimenzionalnom prostoru
- Brzinu u trodimenzionalnom prostoru
- Silu koja djeluje na njega

Primjenom sile na čvorove (točnije na mase čvorova) dolazi do promjene brzine čestice, a time i do promjene pozicije čvora u trodimenzionalnom prostoru. Opruga sama po sebi nema mase, ali ima konstante koje određuju njeno ponašanje. To su:

- konstanta elastičnosti,
- udaljenost između čvorova,
- unutarnje trenje

Iako sama nema masu, opruga direktno utječe na mase na svojim krajevima elastičnom silom te unutarnjom silom trenja.

### 3.2 Prikaz sila koje djeluju u modelu

Kao što je navedeno, opruga utječe na mase elastičnom silom i unutarnjom silom trenja. Elastična sila je definirana kao:

$$F_{elastična} = -k * x$$

gdje su  $F$  elastična sila,  $k$  konstanta elastičnosti opruge, a  $x$  je produljenje opruge od ravnotežnog položaja u sustavu opruge (elongacija opruge). Jednadžba (1) ne zadovoljava naše potrebe utoliko što u ovom slučaju sila uvijek postoji osim kad su čvorovi u istom položaju. S obzirom da čvorovi imaju masu, prema zakonu superpozicije, mase ne mogu nikada biti na istom mjestu (tj. poziciji) u isto vrijeme. To znači da bi sila trebala uvijek postojati. Tu nastaje problem jer će takva sila konstantno utjecati na čvorove (odnosne na mase čvorova) te će doći do neželjenih posljedica (naboravanje opruge). Problem je rješiv uvođenjem konstantne duljine između dva čvora. Tada definiramo elastičnu silu kao:

$$F_{elastična} = -k * (x - d)$$

gdje su  $F$  elastična sila,  $k$  konstanta opruge,  $x$  udaljenost od drugog čvora u sustavu opruge, a  $d$  konstantna udaljenost između čvorova. U ovom slučaju, ukoliko je udaljenost između čvorova  $x$  jednaka konstantnoj udaljenosti  $d$  sila je jednaka nuli. Na taj način rješavamo problem naboravanja.

Udaljenost  $x$  se može dobiti kao razlika pozicija prvog i drugog čvora opruge:

$$x = \text{pozicija}_{\text{čvor1}} - \text{pozicija}_{\text{čvor2}}$$

Time je ujedno dobiven i vektor između dva čvora (tj. dvije mase). Umjesto  $x$  će u daljnjem tekstu biti upotrebljavana varijabla *vektorIzmeđuČvorova*. Sad se može pisati:

$$F_{\text{elastična}} = -k * (\text{duljina}_{\text{vektorIzmeđuČvorova}} - d)$$

Sila djeluje u pravcu vektora *vektorIzmeđuČvorova* te ju je potrebno pomnožiti s jediničnim vektorom kako bi se dobio ispravan izraz:

$$F_{\text{elastična}} = -k * (\text{duljina}_{\text{vektorIzmeđuČvorova}} - d) * \left( \frac{\text{vektorIzmeđuČvorova}}{\text{duljina}_{\text{vektorIzmeđuČvorova}}} \right)$$

Osim elastične sile, u opruzi djeluje i unutarnja sila trenja. Primjer za ovu silu je najočitiiji kad se jedan od čvorova giba. S obzirom da je spojen za drugi čvor, dolazi do usporavanja zbog djelovanja sile drugog čvora. Unutarnju silu trenja stoga možemo napisati u obliku:

$$F_{\text{trenjeOpruge}} = -k * v$$

gdje je  $k$  konstanta trenja opruge, a  $v$  je vektor kretanja čvorova. Brzinu možemo pisati i kao:

$$v = v_{\text{čvor1}} - v_{\text{čvor2}}$$

gdje su  $v_{\text{čvor1}}$  i  $v_{\text{čvor2}}$  brzine pojedinih čvorova.

Sada se ukupna sila kojom opruga djeluje na čvorove može opisati kao:

$$F_{ukupno} = F_{elastična} + F_{trenjeOpruge}$$

Osim sile kojom opruga djeluje, na čvorove mogu utjecati i razne druge sile poput gravitacijske sile, sile trenja zraka, vjetar itd. Radi jednostavnosti prikaza uzeti ćemo u obzir samo silu gravitacije:

$$G = m_{\check{c}vor} * g$$

gdje je  $G$  sila gravitacije,  $m_{\check{c}vor}$  masa pojedinog čvora, a  $g$  gravitacijska konstanta. Sve ostale sile koje djeluju, a želimo ih simulirati se sad lako mogu dodati zbroju gore navedenih sila. Zbog toga je model čestica povezanih oprugama relativno jednostavan za implementaciju u slučaju većeg broja vanjskih sila koje djeluju.

### 3.3 Utjecaj sila na brzinu i položaj čvorova

Djelovanje sile na masu čvora daju čvoru određenu brzinu. Brzinu možemo prikazati kao:

$$v_{\check{c}vor} = v_0 + \frac{F_{ukupno}}{m_{\check{c}vor}} * \Delta t$$

gdje je  $v_0$  početna brzina čvora u nekom trenutku  $v_{\check{c}vor}$  brzina čvora,  $F_{ukupno}$  ukupna sila koja djeluje na čvor,  $m_{\check{c}vor}$  masa čvora, a  $\Delta t$  vrijeme u kojem je sila djelovala na čvor. Promjenom brzine čvora mijenja se i pozicija čvora u prostoru:

$$pozicija = početnaPozicija + v_{\check{c}vor} * \Delta t$$

gdje je *početnaPozicija* pozicija čvora u prostoru prije nego što je dobio brzinu,  $v_{\check{c}vor}$  brzina čvora, a  $\Delta t$  vrijeme u kojem brzina djeluje na čvor. Ovim dvjema jednadžbama se može relativno precizno opisati gibanje čvora kad na njega



djeluje neka sila te će one biti upotrijebljene pri određivanju položaja čvorova u prostoru.

## **4. Programsko rješenje modela čestica povezanih oprugama**

### **4.1 Upotrijebljene tehnologije**

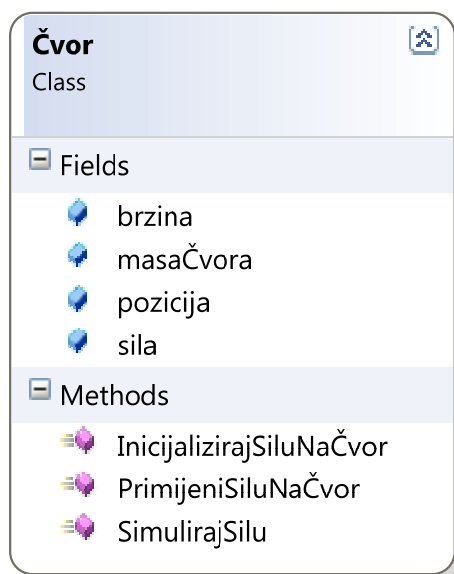
Programsko rješenje ovog rada implementirano je u programskom jeziku C# 3.0 pomoću Microsoftovog Visual Studia 2008., a isprobano je na operacijskom sustavu Windows Vista s ugrađenim Service Packom 1. Za sva iscrtavanja korišten je Microsoftov XNA Game Studio 3.0.

Microsoft XNA Game Studio je niz alata (framework) koji služe za izgradnju interaktivnih video igara. XNA je predstavljen 2004. godine, a najnovija 3.0 verzija je izašla u listopadu 2008. godine. XNA je napravljen od strane Microsofta kako bi oslobodio programere video igara od pisanja kôda koji se ponavlja. Tako svi objekti koji se mogu stvoriti u XNA-u već u sebi sadrže predefiniran kôd koji onda programeri upotpunjavaju svojim idejama. Video igre mogu biti pisane u bilo kojem jeziku koji podržava .NET framework, ali je službeno jezik za XNA C#. XNA osim iscrtavanja ima mnogo više mogućnosti poput podrške za zvuk ili mrežnu komunikaciju te je kompatibilan s raznim fizikalnim i grafičkim pogonima.

Za grafičko sučelje upotrijebljen je grafički podsistem .NET Frameworka 3.0 koji se zove Windows Presentation Foundation (WPF). Sučelje se koristi kako bi se zadali parametri potrebni za simulaciju kao i kako bi se pokrenula sama simulacija. Funkcije iscrtavanja grafičkog sučelja te modela neće biti obrađene u ovom radu.

## 4.2 Programska implementacija čestice

Čestica je modelirana posebnom klasom nazvanom *Čvor*. Unutar klase *Čvor* nalaze se četiri članske varijable te tri članske metode (slika 12.).



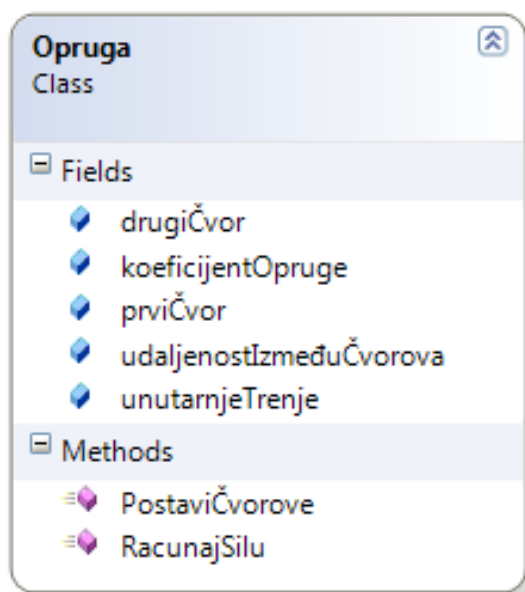
Slika 12. Dijagram klase *Čvor*.

Kao što je opisano u teoretskom dijelu, čvor je predstavljen s četiri vrijednosti: brzinom, masom čvora, pozicijom i silom. Masa čvora je programski predstavljena kao varijabla tipa *float*, dok su brzina, pozicija i sila predstavljene kao varijable tipa *Vector3*. *Vector3* je programska struktura koju XNA koristi kako bi prikazao trodimenzionalne vektore.

Metoda *InicijalizirajSiluNaČvor()* postavlja vrijednost varijable *sila* na nula. Metoda *PrimijeniSiluNaČvor(Vector3 sila)* kao ulazni parametar prima trodimenzionalni vektor te zbraja člansku varijablu *sila* s tim vektorom. Kroz ovu metodu se zbrajaju sve sile koje djeluju na čvor. Metoda *SimulirajSilu(GameTime gameTime)* prima kao parametar varijablu *gameTime* tipa *GameTime*. Struktura *GameTime* je još jedna struktura unutar XNA – ovog seta alata, a sadrži mnoge varijable vezane uz vrijeme izvođenja programa. U ovoj metodi se sila primjenjuje na brzinu koja zatim uzrokuje gibanje čvora.

### 4.3 Programska implementacija opruge

Opruga je slično kao i čestica implementirana posebnom klasom nazvanom *Opruga* (slika 13.).

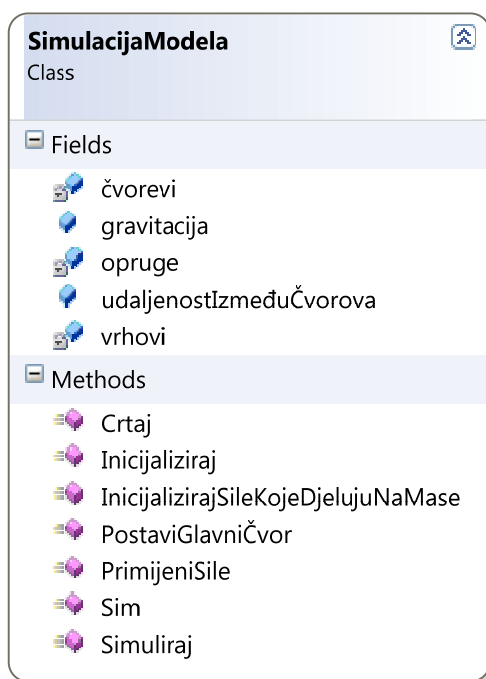


Slika 13. Dijagram klase *Opruga*.

Klasa *Opruga* sadrži pet članskih varijabli i dvije metode. Varijable *prviČvor* i *drugiČvor* su instance klase *Čvor*, dok su ostale varijable parametri tipa *float* koji se zadaju preko korisničkog sučelja. Metoda *PostaviČvorove*(*Čvor a*, *Čvor b*) kao argumente prima dva čvora te inicijalizira *prviČvor* i *drugiČvor* pomoću njih. U metodi *RacunajSilu()* je realizirano izračunavanje elastične sile opruge kao i unutarnje sile trenja. Nakon izračuna ukupne sile u obliku zbroja tih dviju sila, ona se dodaje svakom od čvorova putem metode *PrimijeniSiluNaČvor()* svake od varijabli *prviČvor* i *drugiČvor*. Na taj način se ukupnoj vanjskoj sili koja djeluje na čvor pridodaju sile opruge.

## 4.4 Simuliranje čestica povezanih oprugama

Za upravljanje simulacijom napravljena je posebna klasa *SimulacijaModela*. Kako bi uspješno simulirali model u stvarnom vremenu, u *Main* funkciji se kreira instanca klase *SimulacijaModela*. Ta klasa sadrži sve potrebne funkcije za simulaciju sila na čvorovima i iscrtavanje istih (slika 14.).



Slika 14. Dijagram klase *SimulacijaModela*.

Definira se lista čvorova kao i lista opruga. Između ostalog, ovdje se postavlja gravitacijska konstanta kao i udaljenost između čvorova. Metoda *Inicijaliziraj(int brojČvorova, int pomak)* stvara čvorove, postavlja poziciju čvorova te svakoj opruzi dodjeljuje dva čvora. *InicijalizirajSileKojeDjelujuNaMase()* postavlja sile koje djeluju na čvorove na nulu, a *PrimijeniSile()* računa gravitacijsku silu za mase svih čvorova te za svaku oprugu poziva njenu člansku funkciju koja računa sile kojima opruga djeluje na čvorove. Metoda *Simuliraj()* za svaki od čvorova poziva njegovu člansku funkciju *SimulirajSilu()*. Metoda *Crtaj()* iscrtava cijeli model opruga i čvorova, dok *PostaviGlavniČvor()* postavlja poziciju prvog nepomičnog čvora. *Sim(GameTime gameTime)* je glavna metoda ove klase. Ona prvo pozivom

*InicijalizirajSileKojeDjelujuNaMase()* postavlja sve sile na nulu, zatim računa sve sile koje djeluju na čvor te na kraju ih primjenjuje. U glavnom programu se prvo poziva *Inicijaliziraj()* za sve čvorove, zatim *Sim()* kako bi se simulirale sve sile i promijenile pozicije, te na kraju se poziva *Crtaj()* kako bi se iscrtale nove pozicije čvorova.

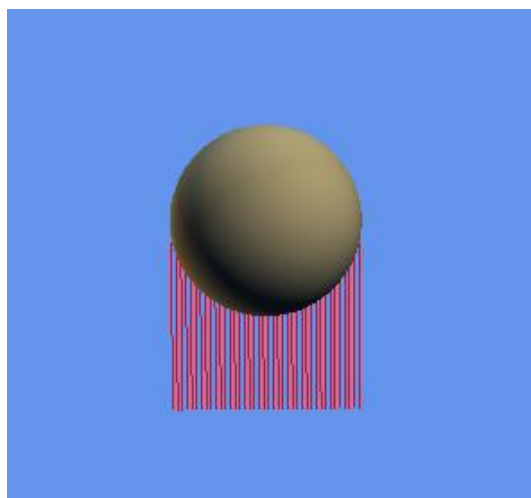
#### **4.5. Krajnji rezultat**

Simulacija ovog modela je napravljena na računalu koje posjeduje dvojezgreni Intelov procesor od 2.2 GHz, grafičku karticu GeForce 8600M GT s memorijom od 512 MB te 3 GB RAM memorije. Simulacija se pokazala samo teorijski ispravna. Nakon programske implementacije, došlo je do niza problema u ponašanju opruge. Najveći je bio konstantno titranje opruge. Prigušenjem brzine je donekle ispravljen rezultat, no također i usporeno kretanje. Daljnjim podešavanjem pojedinih parametara poput konstante elastičnosti opruge, gravitacijske konstante i duljine između dva čvora se dobio koliko toliko ispravan i realan model opruge. Međutim, to je zahtijevalo više računalnih resursa, pa je simulacija bila usporena. Smanjenjem ukupnog broja vlasi te broja opruga na svakoj od njih se postiglo ubrzanje preostalih modela.

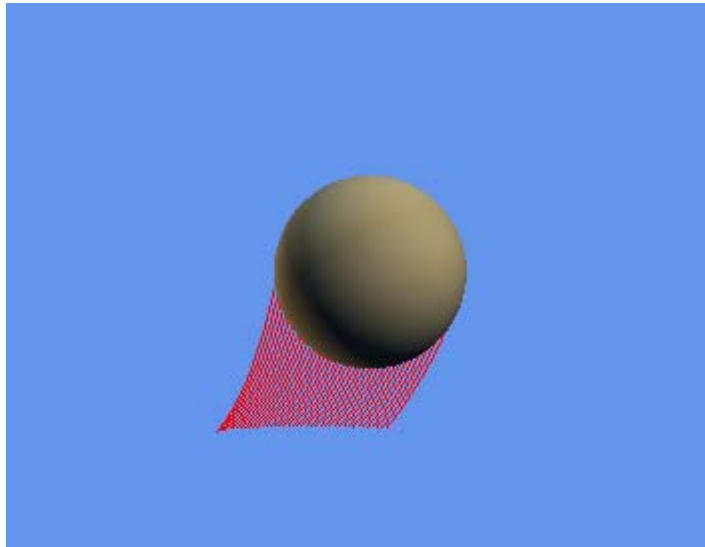
## 5. Zaključak

Sve bržim razvojem računalne animacije počinje se težiti realističnijim modelima svijeta oko nas. Simuliranje vlaknastih objekata je danas dobro razrađeno i koriste se razne tehnike koje daju zadovoljavajuće rezultate. Nažalost, najbolje tehnike nisu popraćene zadovoljavajućom razinom tehnologije, te zbog toga su potrebni razni ustupci kojima se smanjuje realističnost simulacije. Model super zavojnica se pokazao kao najbolji model dosad, ali zbog velike složenosti nije upotrebljiv za simulacije u stvarnom vremenu. Model čestica povezanih oprugama je manje zahtjevan od modela super zavojnica, ali i manje realističan. Također, premda jednostavniji, model čestica povezanih zavojnicama se isto pokazao nezahvalan za simuliranje većeg broja vlasi u stvarnom vremenu.

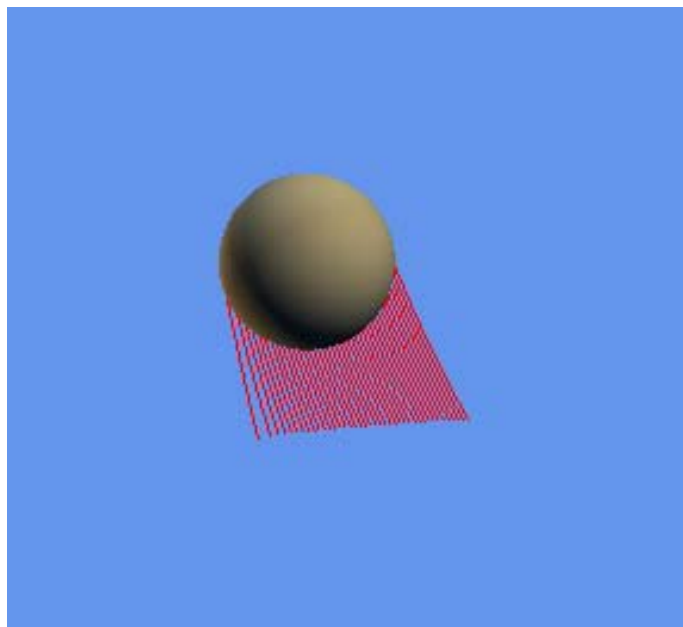
Cilj ovog rada je bila simulacija pojednostavljenog modela čestica povezanih oprugama. Premda teorijski jednostavan, konačan model se ipak pokazao složeniji, te je bilo potrebno uključiti različita prigušenja kako bi se postigao relativno dobar rezultat. Program je pisan u programskom jeziku C#, a ostvaren uporabom XNA frameworka koji se pokazao kao dobro rješenje za iscrtavanje. Konačni rezultat je bio zadovoljavajuć no za realističniju simulaciju potrebno je bolje proučiti svojstva kose kao i drugih parametara koje utječu na simulaciju iste.



Slika 15. Izgled prvih verzija modela.



Slika 16. Izgled modela u pokretu.



Slika 17. Izgleda modela uz djelovanje vanjske sile.



## 6. Popis slika

- Slika 1. Primjena simulacije vlaknastih objekata u filmskoj industriji.
- Slika 2. Primjena simulacije vlaknastih objekata u industriji video igara.
- Slika 3. Modeliranje poligonima.
- Slika 4. Prikaz modeliranja trakama poligona.
- Slika 5. Obično postavljanje 2D slike i Billboard tehnika.
- Slika 6. Koncept čestica povezanih oprugama.
- Slika 7. Model slojeva tekstura.
- Slika 8. Rješenje problema modela slojeva tekstura.
- Slika 9. Vlas sastavljena od 5 motki.
- Slika 10. Usporedba realnog ponašanja kose i ponašanja u modelu super zavojnica.
- Slika 11. Pojednostavljeni prikaz modela čestica povezanih oprugama.
- Slika 12. Dijagram klase *Čvor*.
- Slika 13. Dijagram klase *Opruga*.
- Slika 14. Dijagram klase *SimulacijaModela*.
- Slika 15. Izgled prvih verzija modela.
- Slika 16. Izgled modela u pokretu.
- Slika 17. Izgleda modela uz djelovanje vanjske sile.

## 7. Literatura

- [1] G. Sheppard, sheppard, listopad 2005., *Real-Time Rendering Of Fur*, [www.gamecareerguide.com/education/theses/20051028/sheppard.pdf](http://www.gamecareerguide.com/education/theses/20051028/sheppard.pdf), lipanj 2009.
- [2] J. Lengyel, E. Praun, A. Finkelstein, H. Hoppe, listopad 2008., *Real-Time Fur Over Arbitrary Surfaces*, [research.microsoft.com/en-us/um/people/hoppe/fur.pdf](http://research.microsoft.com/en-us/um/people/hoppe/fur.pdf), lipanj 2009.
- [3] K. Ward, F. Bertails, T. Kim, S. R. Marschner, M. Cani, M.C. Lin, surveyHair, studeni 2006., *A Survey On Hair Modeling: Styling, Simulation and Rendering*, [www-evasion.imag.fr/Publications/2007/WBKMCL07/surveyHair.pdf](http://www-evasion.imag.fr/Publications/2007/WBKMCL07/surveyHair.pdf), lipanj 2009.
- [4] H. Taskiran, U. Gdkbay, prosinac 2004., *Physically-based Simulation of Hair Strips in Real-Time*, [wscg.zcu.cz/wscg2005/Papers\\_2005/Short/A37-full.pdf](http://wscg.zcu.cz/wscg2005/Papers_2005/Short/A37-full.pdf), lipanj 2009.
- [5] S. Sullivan, travanj 2009., *Interactive Hair Simulation Using 2D Strips*, [www.cs.rpi.edu/~cutler/classes/advancedgraphics/S09/final\\_projects/sullivan.pdf](http://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S09/final_projects/sullivan.pdf), lipanj 2009.
- [6] F. Bertails, B. Audoly, M. Cani, B. Querleux, F. Leroy, J. Leveque, srpanj 2006., *Super-Helices for Predicting the Dynamics of Natural Hair*, [www-evasion.imag.fr/Publications/2006/BACQLL06/sigFinalHair06.pdf](http://www-evasion.imag.fr/Publications/2006/BACQLL06/sigFinalHair06.pdf), lipanj 2009.
- [7] A. Selle, M. Lentine, R. Feldkiw, svibanj 2008., *A Mass Spring Model for Hair Simulation*, [physbam.stanford.edu/~aselle/papers/9/hair.pdf](http://physbam.stanford.edu/~aselle/papers/9/hair.pdf), lipanj 2009.
- [8] T. Kim, U. Neumann, prosinac 2001., *A Thin Shell Volume for Modeling Human Hair*, [graphics.usc.edu/cgit/pdf/papers/ca2000.pdf](http://graphics.usc.edu/cgit/pdf/papers/ca2000.pdf), lipanj 2009.

## 8. Sažetak

### Postupci simulacije vlaknastih objekata

Mnogo je postupaka simulacije vlaknastih objekata. Najpoznatije metode simulacije su metoda poligona i tekstura, metoda čestica povezanih oprugama i metoda slojeva tekstura. Najzastupljenija metoda je metoda poligona zbog svoje jednostavnosti i male potrošnje računalnih resursa, a najrealističnija metoda je metoda super zavojnica. Metoda čestica povezanih oprugama postaje sve popularniji način za simulaciju kose. Zasnovana na Newtonovim zakonima dinamike, ova metoda daje relativno dobre rezultate uz neka ograničenja. U ovom radu objašnjen je jednostavan teoretski model ove metode uz programsku implementaciju u jeziku C#. Sile koje su primijenjene na model su sila gravitacije, elastična sila opruge i unutarnja sila trenja opruge.

Ključne riječi: simulacija vlaknastih objekata, simulacija kose u stvarnom vremenu, model čestica i opruga

#### Abstract

### Fibre-simulation techniques

There are many fibre-simulation techniques today. Common ones are surface modelling, mass spring model and thin shell volume. The most used method is surface modelling . It is popular for it's simplicity and low resource consumption. The most realistic method is super-helices method. It's main problem is its high resource consumption. Mass spring model fast becomes one of the most used methods for hair simulation. Based on Newton's Laws of motion, the outcome od this mehod is quite satisfactory though it has some constraints. This paper focuses on the theoretical model of this method. It also implements the method in C#. Only forces applied are the force of gravity, elastic force and friction force.

Keywords: fibre-simulation techniques, real time hair simulation, mass-spring model