

Operația de tăiere în sistemele de particule

Sorina Perșa
Universitatea Tehnică Cluj-Napoca
Catedra de Calculatoare
Barițiu 28
persasorina@yahoo.com

Dorian Gorgan
Universitatea Tehnică Cluj-Napoca
Catedra de Calculatoare
Barițiu 28
Dorian.Gorgan@cs.utcluj.ro

REZUMAT

Lucrarea de față reprezintă o contribuție la crearea mediilor interactive pentru realizarea design-ului vestimentar, prin extinderea unui motor grafic bazat pe un sistem de particule care permite simularea țesăturilor. Problemele tratate în această lucrare se concentrează asupra operației de tăiere, câteva dintre subiectele abordate fiind modalitățile de efectuare a operației de tăiere a țesăturilor, tehnicile de interacțiune utilizabile în procesul de tăiere și implicațiile acestora precum și modelarea procesului de distrugere a constrângerii dintre două particule în operația de tăiere.

Categorii și descriptori ai subiectelor

1.6.3 [Simulation and Modeling]: Applications.

Termeni generali

Design, Experimentation, Performance.

Cuvinte-cheie

Sistem de particule, tăiere, simulare, constrângere, interacțiune.

1. INTRODUCERE

Motoarele grafice bazate pe sistemele de particule au o gamă largă de aplicații. Chiar dacă cea mai populară dintre ele este reprezentată de jocurile pe calculator (în special la nivelul realizării efectelor speciale), de o deosebită importanță sunt și cele din domeniul medical [1] și nu în ultimul rând cele din domeniul design-ului vestimentar [6].

Realizarea cât mai fidelă a unui atelier de croitorie virtual implică pe lângă vizualizare și simulare (de elemente specifice: falduri, cădere liberă, elasticitate, duritate, grosime, coliziune cu alte obiecte), operații asupra țesăturilor, una din principalele operații fiind cea de tăiere.

Sistemele existente orientate către simularea de țesături variază în principal în următoarele aspecte: reprezentarea țesăturilor (rețele regulate sau neregulate [5], [6], cu elementul de bază triunghi sau patrulater), tehnicile de interacțiune folosite în diferitele operații implementate (influențate în primul rând de dispozitivul periferic utilizat: bidimensional sau tridimensional, adică mouse, *tablet*, respectiv *glove*, *stylus*), modalitățile de modificare

interactivă a obiectelor (în două etape: marcarea unui contur pe obiect, urmată de tăierea efectivă a obiectului [1], [7], sau într-o singură etapă, prin distrugerea instantanee a legăturilor [6]).

Până în momentul de față nu s-a găsit o soluție care să fie eficientă, rapidă și precisă și care totodată să ofere utilizatorului controlul ușor asupra operațiilor. Lucrarea de față reprezintă o nouă încercare în această direcție. Plecând de la implementarea unui motor grafic și realizând o trecere în revistă a opțiunilor, lucrarea încearcă să aleagă o soluție care să fie în primul rând intuitivă pentru utilizatorul uman și care să maximizeze performanțele.

Articolul este structurat în trei secțiuni. Secțiunea 2 cuprinde o prezentare de ansamblu a motorului grafic utilizat. Secțiunea 3 descrie operația de tăiere în detaliu, subliniind opțiunile avute în momentul dezvoltării și soluțiile alese spre dezvoltare. Concluziile sunt expuse în secțiunea 4.

2. MOTORUL GRAFIC

Motorul grafic utilizat a fost realizat în C++, utilizând biblioteca grafică OpenGL. Motorul are ca și componentă logică centrală sistemul de particule, care este o colecție de obiecte (reprezentate ca o mulțime de particule unite prin intermediul unor legături) asupra cărora se impun o serie de reguli. Sistemul este alcătuit din următoarele elemente structurale: particule, extensii de particule, constrângeri globale (trebuie respectate de toate particulele sistemului), constrângeri simple (se aplică unei submulțimi distincte de particule din colecția sistemului) și efecte. Manipularea elementelor sale componente este realizată prin intermediul modelului matematic care poartă denumirea de integrarea Verlet [9]. Acesta tratează coliziunile prin proiecție, rezolvă constrângerile prin relaxare și modelează corpurile rigide prin particule cu constrângeri. Funcționarea sistemului de particule presupune executarea unei secvențe de pași. În ordine cronologică, aceasta constă în actualizarea sistemului de particule și a componentelor sale o dată cu trecerea timpului, urmată de desenarea sistemului în noua stare, actualizată.

Motorul grafic curent constituie punctul de plecare al unui atelier de design virtual. Obiectele geometrice regulate (sferă, cub, octaedru) ar putea fi utilizate la crearea unui manechin virtual, iar obiecte precum *mesh* (rețea, plasă)

sau *chain* (lanț) ar putea simula țesăturile necesare la crearea unor vestimentații pentru acesta.

3. TĂIEREA ȚESĂTURII

3.1 Tehnici de tăiere

Tăierea țesăturilor în scopul realizării design-ului vestimentar, se poate efectua în două moduri [2]: fie prin desenarea unei forme bidimensionale pe țesătură, operație urmată de discretizarea acesteia, fie prin selectarea marginii formei dorite, pe un material care este deja discretizat.

A doua soluție este un proces mai apropiat de operația de tăiere din lumea reală, permite menținerea structurii regulate a țesăturii și poate fi implementată mult mai ușor. Motivele menționate anterior, la care se adaugă faptul că se pornește de la o implementare discretizată a țesăturii, fac ca cea de a doua opțiune să fie cea pe care am ales-o în procesul de dezvoltare.

Procesul de tăiere poate fi realizat direct, concomitent cu mișcarea obiectului care realizează operația (foarfecă, cuțit, etc.) sau poate fi simulat prin crearea unui tipar pe suprafața materialului și distrugerea concomitentă a tuturor legăturilor existente între particulele aflate la granița dintre cele două regiuni. Cea de a doua variantă ar putea fi asemuită cu operația automată de tăiere din atelierele de croitorie (vezi Figura 1), dar nu este o operație efectuată în timp real.

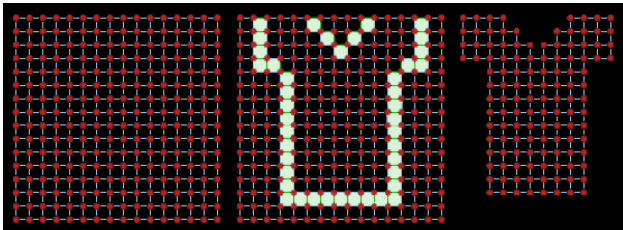


Figura 1. Tăierea prin crearea unui tipar

Când se realizează procesul de tăiere, extragerea formei dorite se poate efectua prin selectarea marginii tiparului. Există totodată și alte opțiuni cum ar fi extragerea unor părți sau crearea unor găuri pe bucata de material.

3.2 Tehnici de interacțiune

Tehnicile de interacțiune care s-ar potrivi operației de tăiere sunt simularea unui cuțit [4], reprezentat ca un obiect bidimensional (grosimea acestuia, care reprezintă cea de a treia dimensiune, este ignorată) într-o scenă tridimensională, sau folosirea cursorului dispozitivului mouse.

3.2.1 Cuțitul virtual

Cuțitul virtual pare să modeleze mai bine lumea reală. Dar în acest caz, suprafața care realizează propriu-zis tăierea nu este un punct, ci un segment de dreaptă; aceasta implică distrugerea concomitentă a mai multor legături dintre particule. În situațiile reale, când tăierea se realizează

manual, fără desenarea în prealabil a unui tipar, obiectul folosit este de obicei o foarfecă și implică distrugerea la un moment dat a unei singure constrângeri.

3.2.2 Cursorul dispozitivului mouse

Dispozitivul mouse este alegerea cea mai des întâlnită în majoritatea operațiilor din sistemele existente. Cursorul permite distrugerea unei singure legături dintre două particule la un moment dat. Problema majoră în cazul dispozitivului mouse este că permite doar o mișcare într-un spațiu bidimensional, iar noi avem nevoie de o mișcare tridimensională. La aceasta se adaugă modalitatea de mapare din spațiul ecran în spațiul virtual, mapare care trebuie să se facă diferențiat, în funcție de poziția cursorului în cea de a treia dimensiune.

Simularea mișcării cursorului mouse în cea de a treia dimensiune s-ar putea realiza prin intermediul rolei (*scroll*) dispozitivului mouse. O vizualizare a acestei mișcări ar implica crearea unei noi vederi a scenei. Noua vedere ar putea fi obținută prin rotirea întregii scene cu 90° , în jurul axei Oy .

Implementarea acestei soluții a întâmpinat următoarele obstacole: nu toate sistemele sunt dotate cu mouse cu scroll și GLUT API [8], pe baza căruia este construit motorul grafic, nu are funcții care să permită utilizarea rolei dispozitivului mouse. Din motive de portabilitate nu s-a dorit utilizarea altei interfețe programator pentru captarea evenimentelor cursorului de la utilizator. Totodată, maparea coordonatelor cursorului din spațiul ecran în spațiul virtual ar fi trebuit să se realizeze diferențiat și în funcție de dimensiunea ferestrei.

Luând în considerare faptul că țesăturile sunt suprafețe, se poate realiza o mapare a punctelor din sistemul de coordonate al cursorului pe particulele țesăturii [1]. Această abordare echivalează cu ancorarea dispozitivului de tăiere de suprafața țesăturii. Dezavantajul major al acestei abordări este restrângerea spațiului de acțiune al dispozitivului la ultima componentă conexasă a țesăturii.

3.2.3 Soluția hibridă: cursorul virtual

Plecând de la tehnicile prezentate anterior, care prezintă fiecare avantaje și dezavantaje ei, s-a ajuns la concluzia că soluția cea mai bună ar fi o combinație a celor două. Pentru aceasta se simulează cursorul mouse printr-un obiect reprezentat în scenă sub forma unei sfere mici, de dimensiunea cursorului. Această sferă va fi *cutter*-ul, care va fi modelat punctiform în operația de tăiere.

Translația în planul XY se efectuează prin intermediul mișcărilor dispozitivului mouse, după activarea în prealabil a unui buton din fereastra aplicației. Pe același principiu se bazează și mișcarea pe axa Oz, dar este necesară activarea altui buton.

3.3 Aproximarea operației

Vom analiza tăierea unei legături, ignorând

dimensionalitatea spațiului în care lucrăm. O legătură este reprezentată vizual ca un segment de dreaptă, care unește două particule. În cazul real, o legătură este distrusă în momentul în care foarfeca sau dispozitivul de tăiere utilizat intră efectiv în contact cu respectiva legătură. În cazul în care dispozitivul de tăiere este modelat punctiform, pentru ca cele menționate anterior să fie aplicabile, poziția dispozitivului trebuie să aparțină efectiv legăturii. Operând într-un mediu discret, acest lucru este posibil, dar totuși dificil de realizat pentru utilizatorul uman. Ca urmare, se va folosi o aproximare: distanța de la punctul de referință al tăierii la legătură trebuie să fie mai mică decât un prag stabilit în prealabil pentru a putea afirma că distrugerea legăturii respective a fost realizată cu succes. Acest prag este determinat luând în considerare atât îndemânarea utilizatorului uman cât și gradul de precizie necesar operației care se execută.

Rezultatul operației este influențat totodată de modul de calculare al distanței de la un punct la un segment de dreaptă. Cazul în care perpendiculara din punctul de tăiere pe dreaptă aparține segmentului este tratat identic în ambele moduri. Diferența apare când această perpendiculară se situează în exteriorul segmentului: o modalitate ar fi considerarea distanței ca infinită, iar a doua ar fi calcularea distanței de la dispozitivul de tăiere până la punctul cel mai apropiat de acesta, care coincide cu unul din capetele segmentului. Numărul de legături distruse este unu în cazul distanței infinite și poate ajunge până la gradul de conectivitate al particulei care reprezintă capătul segmentului în cel de al doilea caz, dacă distanța calculată este mai mică decât pragul ales. În soluția propusă, utilizatorul poate alege modalitatea de calculare.

3.4 Consecințele operației de tăiere

Modelarea distrugerii legăturii dintre două particule se poate realiza fie prin simpla distrugere a constrângerii, fie prin distrugerea constrângerii urmată de crearea a două particule în punctul de tăiere, la care se adaugă crearea constrângerilor corespunzătoare integrării noilor particule.

În modul ideal, alegerea opțiunii s-ar face adaptiv, la rulare, în funcție de densitatea de particule. Astfel, în cazul în care avem de a face cu o densitate mare de particule, respectiv distanța dintre două particule este mică, se poate folosi simpla distrugere a constrângerii, iar în cazul în care avem de a face cu o densitate mică de particule, se poate utiliza cea de a doua variantă.

Pentru lanțul de particule, reprezentat ca un tablou unidimensional, crearea unor noi particule și inserarea lor în tablou se poate realiza facil, așadar alegerea opțiunii se poate face într-adevăr adaptiv. Crearea noilor particule se poate face la mijlocul legăturii distruse sau chiar în punctul de tăiere. Pentru a evita crearea unei densități mari de particule, a fost aleasă prima modalitate. Rezultatul poate fi vizualizat în Figura 2.

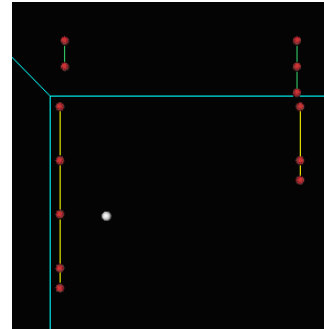


Figura 2. Rezultate experimentale: distrugerea adaptivă a constrângerilor în cazul lanțului de particule

În cazul rețelei de particule (*mesh*), doar prima opțiune poate fi utilizată, deoarece rețeaua este reprezentată ca o matrice de puncte, unde fiecare particulă are o conectivitate de 4. Introducerea particulelor și a constrângerilor corespunzătoare ar implica conectivitate variabilă (vezi Figura 3).

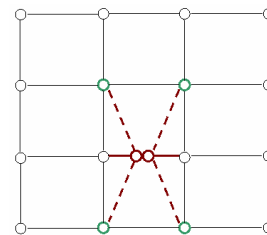


Figura 3. Crearea conectivității variabile

Regularitatea rețelei, necesară pentru o operare ușoară și rapidă asupra elementelor, nu permite introducerea de noi particule după metoda prezentată anterior. Particule noi ar putea fi adăugate doar dacă se renunță la reprezentarea sub forma de matrice a rețelei. Aplicarea texturii pe țesătură este procesul cel mai afectat de mărirea conectivității particulelor.

Operația de tăiere are consecințe și asupra extensiilor de particule. O extensie de particule este o submulțime de particule care poate fi mai ușor manevrată. O țesătură reprezentată sub formă de rețea formează o extensie de particule. În momentul în care realizăm tăierea unui material putem ajunge în situația în care avem două submulțimi disjuncte de particule. Există două opțiuni: cele două submulțimi să fie privite în continuare ca o singură extensie, sau o nouă extensie să fie creată pentru una din submulțimi. Dacă tăierea are ca scop realizarea unor forme, crearea de noi extensii este foarte utilă, deoarece permite o ușoară eliminare a resturilor (vezi Figura 1). Pe de altă parte, această operație ar presupune o verificare continuă a apariției unor submulțimi disjuncte ale mulțimii inițiale, ceea ce ar conduce la scăderea vitezei de procesare.

În cazul de față, fiecare extensie este caracterizată de un

nume, care o identifică unic. Selecția obiectului asupra căruia dorim să aplicăm diferite operații (de exemplu tăierea) se face prin selectarea numelui extensiei din lista cu obiectele existente în imagine. Când țesătura inițială este formată din NxM particule, se pot obține până la NxM extensii. Dacă pentru fiecare extensie trebuie creat un nume pe care să îl adăugăm în lista de obiecte, utilizatorul uman va fi copleșit. De aceea se preferă menținerea particulelor în cadrul aceleiași extensii.

3.5 Netezirea materialului

Dacă densitatea de particule din țesătură este destul de mică și se dorește realizarea unei forme care să fie folosită pe urmă la crearea unei haine, în unele cazuri este necesară netezirea marginilor materialului (vezi Figura 4).

[2] propune o soluție pentru netezirea marginilor zimțate. Se selectează particulele de pe marginea care necesită netezire; aceste puncte vor reprezenta puncte de control. Pe baza acestor puncte se calculează curba Bezier de interpolare, iar particulele de pe margine sunt mutate pe această curbă. Curbele Bezier sunt preferate deoarece sunt ușor de implementat și sunt niște instrumente puternice pentru desenarea curbilor. Din păcate, netezirea unei forme nu poate fi automatizată deoarece nu toate marginile necesită netezire.

Netezirea este mult mai ușor de realizat în sistemele de particule unde se lucrează cu conectivitate de 8, sau în sistemele care se bazează pe o rețea triunghiulară neregulată [5]. Rețelele neregulate permit modelarea unor forme mai complicate, care pot conține curburi cu un număr scăzut de particule. Dezavantajul conectivității de 8 este timpul de procesare mai mare. Dacă se folosește o țesătură formată din particule cu o densitate foarte mare, netezirea s-ar putea să nu fie necesară.

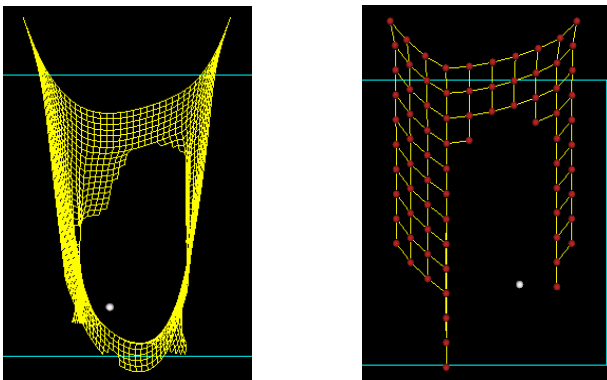


Figura 4. Rezultate experimentale: distrugerea constrângerilor în cazul rețelilor de particule

3.6 Automatizarea tăierii

Una din caracteristicile atelierelor de croitorie este automatizarea operației de tăiere după un șablon prestabilit.

În cazul atelierului virtual, aceasta s-ar putea realiza prin citirea unor fișiere care să conțină descrierea geometriei tiparului [3]. Acest tipar ar fi desenat pe țesătura aleasă, iar tăierea efectivă s-ar realiza prin apășarea unui buton, care ar distruge instantaneu legăturile marcate.

4. CONCLUZII

Lucrarea a prezentat o metodă de tăiere în timp real a țesăturilor virtuale într-un spațiu geometric complex. Utilizând un motor grafic bazat pe integrarea Verlet, metoda a fost implementată în C++, folosind librăria grafică OpenGL și pentru vizualizare GLUT API. Scopul metodei a fost o simulare cât mai fidelă a operației de tăiere reale, eliminând necesitatea definirii în prealabil a unui contur de tăiere. Interacțiunea cu utilizatorul s-a realizat prin intermediul unui obiect reprezentat în spațiul virtual, obiect a cărui mișcare este realizată cu dispozitivul mouse. Această soluție nu necesită echipamente suplimentare, deci este viabilă pentru orice utilizator înzestrat cu un PC. Etapa următoare este realizarea operației de coasere a formelor generate anterior.

5. REFERINȚE

- [1] Bruyns, C. D., Senger, S., Interactive cutting of 3D surface meshes. *Computer&Graphics* 25 (2001), 635-642.
- [2] Durupnar, F., *A 3D garment design and simulation system*, Masters degree Thesis, Bilkent University, Ankara, Turkey, 2004
- [3] Spanlang, B., Vassilev, T.I., *Efficient Cloth Modeling for Dressing Animated Virtually People*, 2001
- [4] Szeliski, T., Tonnesen, D., Surface Modeling with Oriented Particle Systems. *ACM Symp. Computer Graphics (Siggraph)*, 1992, 185-194
- [5] Thalmann, N.M., Volino, P., Developing Simulation Techniques for an Interactive Clothing System. In *Proceedings of the 1997 International Conference on Virtual Systems and Multimedia*, 1997
- [6] Thalmann, N.M., Volio, P., Interactive Cloth Simulation: Problems and Solutions, <http://www.miralab.unige.ch/papers/52.pdf>
- [7] WongKC, Sin TY, Heng P. Interactive volume cutting. In *Proceedings of the Graphics Interface 1998 Conference*, (Vancouver, Canada, June 18-20, 1998)
- [8] * * *, *OpenGL Utility Toolkit*: <http://www.opengl.org/resources/libraries/glut.html>
- [9] * * *, *Verlet Integration*: <http://www.answers.com/topic/verlet-integration>