De unde ştiu melodia asta?

Cristian Frâncu — mailto:francu@cs.rutgers.edu, http://www.cs.rutgers.edu/~francu
Mihai Budiu — mailto:mihaib+@cs.cmu.edu, http://www.cs.cmu.edu/~mihaib/

august 2000

Subiect: recunoaşterea muzicii

Cunoştinţe necesare: cunoştinţe elementare despre muzică şi baze de date

Cuvinte cheie: semnal, eşantionare, cuantizare, căutare, indexare, bibliotecă digitală

Cuprins

1 Evoluţia calculatoarelor spre multimedia

John von Neumann este considerat creatorul arhitecturii calculatoarelor aşa cum o cunoaştem astăzi. Neumann a ajuns la calculatoare din dorinţa de a rezolva probleme de fizica fluidelor. Multe ecuaţii diferenţiale nu admit soluţii analitice (adică sub forma unei formule), ci pot fi rezolvate doar aproximativ, numeric. Cum calculatoarele cu care avea de a face erau extrem de nefiabile, von Neumann şi-a suflecat mînecile şi s-a apucat să le amelioreze, pentru a putea găsi soluţiile care-l interesau. El a propus o mulţime de concepte care şi astăzi se regăsesc în arhitectura calculatoarelor, cum ar fi separarea funcţionalităţii fizice şi logice a circuitelor (adică construcţia de porţi logice din tuburi – mai tîrziu tranzistoare, şi acum circuite integrate, care apoi pot fi manipulate independent de modul în care sunt construite, ca nişte obiecte matematice ideale), şi diviziunea calculatorului în unitate de procesare, unităţi de intrare-ieşire şi memorie în care sunt stocate atît programele cît şi datele prelucrate.

Dar, deşi a anticipat şi dat naştere multor idei seminale, povestea spune că von Neumann a fost foarte supărat pe un colaborator de-al său, care încerca să scrie un asamblor. Asamblorul e un program care uşurează munca programatorului, permiţîndu-i să scrie programele într-un limbaj textual în loc de a folosi şiruri de zero şi unu, în cod-maşină. Von Neumann spunea că timpul de calcul este irosit făcînd o activitate pe care o poate face foarte bine omul.

Astăzi afirmaţia lui von Neumann ni se pare absolut anacronică, dar acest lucru se întîmplă pentru că balanţa între costul puterii de calcul şi cel al muncii omeneşti s-a schimbat absolut dramatic; la ora de astăzi orice investiţie care mută efortul omenesc pe seama unui calculator este productivă.

De-a lungul timpului tipurile de prelucrare făcute de calculatoare s-au diversificat enorm; creşterea capacităţii de stocare şi prelucrare ne-a permis să trecem încetul cu încetul de la procesarea unor cantităţi infime de date la folosirea unor baze de date enorme, la folosirea interfeţelor grafice sofisticate, la comunicaţii care înglobează tot mai mult imagine, sunet, film, culoare.

Dar dacă astăzi lumea strîmbă din nas la ideea folosirii unui browser în mod text (deşi numai în urmă cu 10 ani gopher-ul era cea mai avansată tehnologie), evoluţia în complexitatea informaţiei procesată în mod curent de către calculator este departe de a fi încheiată. Chiar dacă astăzi o mulţime de site-uri de web au grafică spectaculoasă, şi jocurile în spaţii tridimensionale sunt banalităţi, o mulţime de alte operaţii elementare sunt încă foarte greu de făcut de către calculator. Chiar stocarea şi transmisiunea informaţiei video este extrem de primitivă, cele mai evoluate sisteme putînd transmite imagini de dimensiunea unui timbru (vorbim de produse disponibile comercial, şi nu de experimente în laborator).

În ceea ce priveşte imaginile şi muzica, situaţia este ceva mai bună: ştim să le stocăm, livrăm, reproducem şi înregistrăm. Dacă este însă vorba de procesări mai complicate, suntem încă foarte neajutoraţi.

În textul de faţă vom investiga felul în care putem manipula muzica digitală.

Acest articol se bazează pe cercetarea unuia dintre autori, Cristian Francu; rezultatele sale au fost prezentate de curînd cu mult succes la conferinţa internaţională despre multimedia ICME 2000. Sistemul construit de el, care recunoaşte melodii după exemple, este probabil cel mai performant în existenţă la ora actuală.

Algoritmii dezvoltaţi de Cristi împreună cu grupul lui de cercetare vor forma baza unei biblioteci digitale de muzică, în curs de dezvoltare.

2 Despre sunet

Dar înainte de a putea plonja în detaliile modului în care muzica poate fi indexată şi căutată, va trebui să discutăm puţin despre cum se înregistrează muzica în mod digital.

Sunetul se propaga prin aer sub forma unei unde; această undă poate fi caracterizată prin amplitudinea ei în fiecare moment. Un sunet "pur", fundamental, are amplitudinea descrisă de o sinusoidă. Frecvenţa sinusoidei (inversul perioadei) se numeşte "frecvenţa sunetului", şi se măsoară în hertzi, Hz.

Cînd una sau mai multe surse emit mai multe sunete simultan, undele lor se suprapun (se adună) şi auzim sunetul rezultant. Dintre sunetele instrumentelor muzicale, cel mai apropiat de un sunet pur este cel al flautului. Toate instrumentele emit de fapt o mixtură de sunete, dintre care unul este preponderent; de aceea nota "la" cîntată la pian sună altfel decît nota "la" cîntată la vioară, deşi ambele au aceeaşi frecvenţă (stabilită printr-o convenţie internaţională în 1939 la 440hz). În natură nu există surse de sunete pure, dar ele pot fi generate electronic, folosind oscilatoare.

Chiar dacă putem genera sunete pure, urechea omenească nu le poate auzi: chiar organele de simţ din ureche vibrează ele însele şi intră în rezonanţă cu alte frecvenţe decît cele prezente în sunetul ambiant (generează ceea ce se numesc "armonici auriculare"). Pe lîngă faptul că generează sunete inexistente, urechea omenească este inegal de sensibilă la frecvenţe diferite; aparatul auditiv poate percepe sunete între 16hz şi 24khz (desi multi experti vor afirma ca sunete peste 20 khz sunt absolut inaudibile, iar cei mai multi oameni aud pina in jur de 15 khz), fiind cel mai sensibil în jurul frecvenţei de 4khz (sensibil însemnînd că poate auzi semnalele cu energia cea mai mică). De asemenea, sensibilitatea scade cu vîrsta.

Toate aceste slăbiciuni ale aparatului auditiv uman sunt exploatate de metodele de înregistrare digitală. Toate aceste metode *filtrează* sunetul în aşa fel încît să înregistreze doar o bandă îngustă de frecvenţe.

Putem distinge două clase mari de metode de stocare a muzicii sub formă digitală: înregistrările şi descrierile simbolice. Ne vom ocupa pe scurt de fiecare dintre ele.

2.1 Înregistrări digitale

Înregistrările digitale sunt echivalente cu tehnicile de înregistrare analogică patentate de Edison în urmă cu 127 de ani: captează un semnal din aer, îl prelucrează, codifică şi stochează. Spre deosebire de patefon, tehnicile digitale descriu semnalul prin secvenţe de cifre şi nu prin procese fizice (de exemplu, patefonul descrie semnalul prin forma şanţurilor de pe disc).

2.1.1 Eşantionare şi cuantizare

Pentru a putea transforma o mărime continuă într-o serie de numere trebuie să facem două operaţii distincte: să o *eşantionăm* şi să o *cuantizăm*. Figura 1 ilustrează aceste procedee.

Eşantionarea măsoară semnalul audio din timp în timp, de obicei cu o perioadă fixă. Teorema lui Nyquist arată că dacă vrem să înregistrăm un semnal ale cărui frecvenţe componente sunt sub x hz, e suficient să eşantionăm semnalul cu frecvenţa $2x$ hz. Compact-disc-urile de pildă eşantionează sunetul la 44.1 khz, ceea ce este dublul lui 22.05khz, care este o limită mult înafara auzului celor mai multe persoane (dacă nu a tuturor). Teorema lui Nyquist arată că din aceste eşantioane, în

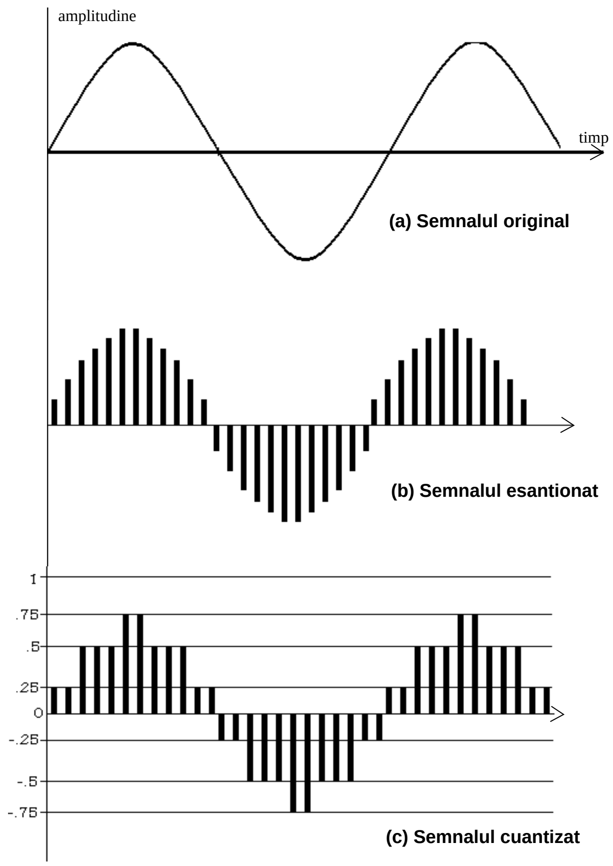

Figura 1: Semnalele înregistrate sunt funcţii continue de timp (a). Semnalul este întîi *eşantionat*, măsurînd valoarea lui într-un număr finit de puncte, de regulă echidistante (b). Apoi valorile eşantioanelor sunt codificate cu un număr finit de biţi în procesul de *cuantizare* (c). În această figură cuantizarea s-a făcut cu 3 biţi, care pot distinge opt valori diferite ale amplitudinii (unele pozitive, unele negative.)

număr finit, se poate reconstitui *complet*, fără nici un fel de pierderi, forma semnalului original, continuu!

Astfel am rezolvat problema transformării unui semnal continuu într-o secvenţă finită de valori discrete. Dar fiecare din aceste valori poate fi exprimată printr-un număr real care ar avea nevoie de o precizie infinită pentru a fi exprimat. Sistemele de înregistrare digitală aproximează valorile posibile ale amplitudinii printr-un număr cu o reprezentare finită; acesta este procesul de *cuantizare*. Intervalul între amplitudinea zero şi o amplitudine maximă posibilă este împărţit în segmente (de obicei egale). Un numar finit de biti este suficient pentru a numerota segmentele. De exemplu, compact disc-urile folosesc 16 biţi de precizie pentru a descrie o valoare a amplitudinii; asta înseamnă că pot distinge $2^{16} = 65536$ de nivele diferite de amplitudine, suficient de fine pentru a păcăli urechea umană.

2.1.2 PCM şi CD-urile

Dacă înregistrăm valoarea eşantioanelor, în ordinea în care apar în semnal, înregistrăm efectiv sunetul. Acest gen de descriere a unui semnal se numeşte "modulare cu pulsuri", *pulse code modulation*, prescurtat PCM.

Compact-disc-urile stochează informaţia necomprimată; dimpotrivă, pentru a preveni erorile care pot apărea din cauza zgîrieturilor, expandează informaţia şi o codifică în mod redundant: dacă unii biţi dispar, ei pot fi reconstituiţi, în măsura în care stricăciunile nu sunt prea mari.

Un CD înregistrat stereo are deci două canale care conţin 44100 de valori de 16 biţi pentru fiecare secundă de muzică; asta înseamnă o rată de transfer a informaţiei de 2 * 44100 * 16 = 1.411Mbps (megabiţi pe secundă). Aceasta este viteza cu care informaţia iese din CD-player-ul calculatorului dumneavoastră; în interior rata este chiar mai mare, din cauza codificării redundante. Un compact disc poate stoca aproximativ 70 de minute de muzică, avînd deci o capacitate de 70 * 60 * 1.411 / 8 = 740 MB (megaocteţi). Pe un hard-disc de 20 de gigaocteţi putem deci stoca aproximativ 25 de CD-uri.

Tipul standard de fişier de sunet în Microsoft Windows este cel cu extensia `.wav`, care codifică de obicei informaţia tot în formă PCM. De asemenea, PCM este metoda folosită pentru codificarea vocii în telefonia digitală şi ISDN.

2.1.3 Compresie digitală

Prin eşantionare şi cuantizare putem stoca informaţiile din orice semnal continuu (în anumite limite de frecvenţă). Adesea însă informaţia din acest semnal este foarte amplă şi redundantă. Depinzînd de mediul de stocare şi transmisie putem dori să reducem informaţia sau nu.

Mai multe scheme din ce în ce mai sofisticate au fost dezvoltate pentru a comprima muzica. Putem distinge două mari categorii de metode de compresie: compresie cu pierderi şi compresie fără pierderi. Compresia fără pierderi poate oricînd reconstitui identic semnalul original; cea cu pierderi decide să arunce la gunoi informaţiile "neimportante".

Compresie fără pierderi Un program care comprimă un semnal audio se numeşte *codec*, de la **co**dor-**dec**odor. Fiecare firmă mare din telecomunicaţii, industria electronică a echipamentelor audio şi din software a dezvoltat propriile codec-uri; cele mai multe dintre acestea sunt secrete bine păzite.

În general, nu putem comprima decît puţin muzica dacă insistăm să nu avem pierderi (lossless), dar pentru aplicaţii specifice compresia fără pierderi este acceptabilă ca performanţă (măsurată prin resurse de calcul consumate pentru a obţine o anumită calitate a sunetului). De exemplu, vocea umană are un spectru relativ îngust, şi variaţiile de intensitate şi frecvenţă nu sunt foarte mari în timp scurt.

Metode eficace de compresie sunt compresia diferenţială (DPCM) şi compresia adaptivă diferenţială (ADPCM). Compresia diferenţială, în loc de a codifica fiecare eşantion separat, codifică diferenţa dintre două eşantioane succesive; dacă semnalul nu poate varia foarte repede, această diferenţă va fi întotdeauna mică, şi deci va putea fi transmisă cu mai puţini biţi (folosind de exemplu un cod Huffmann).

Compresia adaptivă merge chiar mai departe: foloseşte un model predictiv al semnalului, şi calculează în fiecare clipă care este cea mai probabilă valoare a semnalului. Apoi stochează diferenţa între semnalul real şi valoarea prezisă. Dacă predictorul este foarte bun, diferenţa va fi adesea minusculă. Predictorii înşişi sunt adesea adaptivi, învăţînd o serie de coeficienţi de predicţie în timp ce fac prezicerea (adică învaţă din propriile erori). Acelaşi predictor este folosit atît la sursă cît şi la destinaţie.

Compresie cu pierderi Compresia fără pierderi poate cel mult obţine reduceri de 2-3 ori a cantităţii de informaţie dintr-un semnal muzical. De aceea, cele mai folosite pe scară largă la ora actuală sunt compresiile cu pierderi (lossy). Cele mai eficace scheme de compresie cu pierderi se bazează pe sensibilitatea urechii umane pentru a deduce ce părţi din semnal pot fi eliminate, şi de aceea se numesc *compresii perceptuale*.

De pildă, dacă auzim un sunet foarte puternic, alte sunete slabe din imediata vecinătate temporală (cîteva milisecunde) sunt acoperite. Dacă le eliminăm pe acestea din semnal, putem transporta mai puţină informaţie. Sunetele puternice acoperă vecinii lor apropiaţi atît în timp cît şi în frecvenţă; asta înseamnă că un sunet puternic va împiedica percepţia altor sunete de frecvenţe apropiate sau care se aud în imediata vecinătate.

Pentru a descoperi părţile nesemnificative, codec-urile perceptuale adesea fac o *analiză spectrală* a sunetului, extrăgînd frecvenţele componente. Apoi calculează cîtă energie sonoră este în fiecare frecvenţă, şi cele care sunt sub pragurile audibile, sau care sunt acoperite de alte frecvenţe care se aud mai tare, sunt eliminate din semnal.

Alte trucuri deştepte sunt folosite pentru a reduce informaţia cînd înregistrăm canale stereo, sau chiar sound surround (cinci canale): de exemplu se înregistrează doar un canal şi diferenţa de la acesta la celălalt, care tinde să fie mică, şi se folosesc canale cu bandă de frecvenţă redusă pentru başi (subwoofers).

Printre cele mai importante formate de compresie cu pierderi sunt:

Dolby Digital numit şi AC-3, care este folosit în televiziunea de înaltă rezoluţie (HDTV), DVD, discuri laser, televiziune digitală şi în unele cinematografe pentru sunetul filmelor.

MPEG audio este o familie mare de standarde, care cuprinde mai multe variante MPEG1, MPEG2, MPEG4, MPEG7; MPEG1 şi MPEG2 la rîndul lor au sub-nivele 1, 2, 3 şi AAC. MP3 este la ora actuală formatul preferat pentru schimb de muzică pe Internet, datorită compresiei care poate ajunge pînă la 12/1; este o prescurtare de la MPEG 2 level 3.

Liquid Audio este un format sofisticat pentru a reprezenta muzica, bazat atît pe Dolby Digital cît şi pe MPEG AAC. Permite integrarea unor tipuri de date felurite în fluxul muzical (imagini, text, partitura, preţ, versuri, etc.).

Apple quicktime e un format dezvoltat de Apple, care permite codificarea atît a muzicii cît şi a informaţiei video.

DVD Audio este încă în curs de dezvoltare; ar fi trebuit să apară deja, dar fabricanţii încearcă să dezvolte metode eficiente împotriva copierii.

Formate flux (streaming) E o diferenţă foarte mare între a cînta muzică de pe un disc aflat în unitatea calculatorului propriu şi a transfera muzica de la distanţă prin reţea: discul local poate garanta un flux susţinut de informaţii, fără erori şi pierderi. Internet-ul este însă impredictibil, şi nu orice tip de format este potrivit pentru transmisiune.

Dacă mai dorim şi să reproducem informaţia în timp ce este primită avem de-a face cu un format de tip flux (streaming). Formatele flux trebuie să prevină erorile care pot apărea. Două tehnologii sunt folosite pentru transmisiune pe o reţea nefiabilă:

- Stocarea (buffering): în funcţie de variabilitatea (jitter) pachetelor de date care sosesc la receptor, acesta trebuie să întîrzie redarea informaţiei cu cîteva secunde. De pildă, dacă un pachet poate face între 3 şi 9 secunde de la sursă la destinaţie, destinatarul trebuie să cînte cu cel puţin 6 secunde întîrziere. În felul acesta, ştie că atunci cînd vrea să cînte pachetul următor, acesta a sosit deja.

- Codificarea redundantă: pentru că adesea încărcătura serverului şi timpul de transmisiune dus-întors face imposibilă re-transmiterea pachetelor pierdute, transmiţătorul codifică redundant informaţia, în aşa fel încît un pachet pierdut să poată fi reconstituit în întregime din vecinii săi.

Real Audio este probabil cel mai răspîndit format de tip flux în Internet, dar Microsoft împinge tare propriul format secret, **Windows Media Audio**.

2.2 Formate simbolice

Toate formatele pe care le-am trecut în revistă pînă acum sunt destinate pentru a "înregistra" o interpretare. Alte formate, cele simbolice, se mulţumesc să descrie forma echivalentă a *partiturii* muzicale. Acestea sunt deci mult mai compacte, pentru că reprezintă doar notele, duratele lor şi informaţii despre instrumentaţie.

Cel mai răspîndit standard de reprezentare simbolică este *MIDI*, Musical Instruments Digital Interface. Acesta este folosit practic de toate instrumentele electronice pentru codificarea informaţiei.

O metodă hibridă este de a reprezenta notele, ca într-o reprezentare simbolică, dar de a le obţine prin eşantionare. De exemplu, cînd înregistraţi o melodie la o orgă electronică, ea este înregistrată în acest fel, măsurînd la fiecare 2 milisecunde care clape sunt apăsate. MIDI poate conţine multe alte informaţii: mai multe canale separate, informaţii despre ritm, intensitate şi informaţii simbolice despre titlul piesei, autor, etc.

Sistemul pe care îl vom prezenta în continuare foloseşte ca mod de reprezentare a muzicii această din urmă metodă. Dacă melodia conţine o singură voce (şi nu are acompaniament), ea poate fi reprezentată printr-o funcţie, ca în figura 2. Graficul acestei funcţii arată ca o colecţie de segmente orizontale, numite "funcţii treaptă".

3 Căutare multimedia: o problemă nerezolvată

Căutarea de melodii se poate împărţi în cîteva categorii. Prima, şi cea mai populară, este de departe căutarea de cîntece în format MP3, sau alte formate de bună calitate, care sunt destinate ascultării. Acum cîteva luni cererile la motoarele de căutare pentru fişiere MP3 au trecut pe locul întîi ca frecvenţă, depăşind liderul anterior, pornografia. A doua categorie este căutarea de versuri,

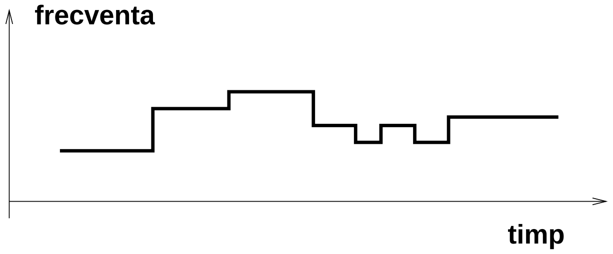

Figura 2: Reprezentarea muzicii sub formă de funcţii-treaptă reprezintă la fiecare moment de timp nota muzicală cîntată în acel moment. Pauzele sunt indicate prin absenţa oricărei note. Timpul este eşantionat cu granularitate mică (în jur de 2 milisecunde).

iar a treia este căutarea de partituri. Obiectele căutate sunt adesea ilegal prezente pe Internet, cei care le oferă încălcînd legea copyright-ului. Sunt ilegale atît publicarea cîntecelor, versurilor, sau partiturilor, cît şi obţinerea lor de pe Internet, fără încuviinţarea posesorului copyright-ului.

Există însă şi un alt fel de căutare, care poate fi perfect legală: identificarea cîntecului. În acest tip de căutare rezultatul este numele cîntecului, eventual autorul, albumul, şi poate o legătură spre un magazin care oferă piesa spre vînzare. De cîte ori vi s-a întîmplat să nu vă puteţi scoate din minte o melodie? Uneori vă amintiţi un vers sau două, caz în care Internetul vă poate ajuta să-l identificaţi. Alteori, tot ce vă mai amintiţi este melodia, caz în care singura speranţă stă în prieteni: poate dacă încercaţi să le cîntaţi un fragment (şi dacă vocea vă permite) şi dacă prietenii recunosc melodia, atunci veţi afla numele cîntecului.

Această ultimă metoda de căutare poartă numele de *căutare după conţinut*; aplicaţiile ei sunt nenumărate. Prin analogie, căutarea după conţinut în domeniul imaginilor[1] găseşte imagini similare pornind de la o imagine-exemplu. Tehnologia căutării după conţinut este încă foarte primitivă pentru tipuri de date multimedia. Să încercăm să vedem care sunt dificultăţile în domeniul muzical.

3.1 Probleme în căutarea după conţinut

Vrem deci ca un utilizator să poată cînta o melodie (întrebarea), şi un sistem electronic să o identifice în mod automat într-o bază de date. Pentru aceasta trebuie să fim capabili să *comparăm* două melodii (sau o melodie cu un fragment — întrebarea), pentru a putea spune cînd două melodii sunt asemănătoare şi cînd sunt diferite. Avem de rezolvat însă mai multe probleme dificile:

Una din problemele cele mai mari este chiar **sursa**. Nu toţi oamenii au ureche muzicală bună şi numai o parte dintre aceştia pot cînta bine. Un sistem care cere o reproducere perfectă a fragmentului este util doar pentru o minoritate restrînsă. De aceea căutarea trebuie sa fie tolerantă la erori în "întrebare": note greşite, note lipsă, note în plus. De asemenea este foarte probabil că fragmentul va fi cîntat mai rapid sau mai lent decît originalul, şi în altă tonalitate.

O a doua problemă este *extragerea notelor* (*pitch tracking*) din întrebare. În urma înregistrării nu obţinem un fişier cu conţinut simbolic (ex. MIDI), ci o înregistrare (ex. `wav`). Căutarea necesită identificarea notelor din acest fişier. Acest proces introduce erori suplimentare: unele note vor ieşi fragmentate, cîteodată note identice vor fi concatenate; vocea umană adesea trece între note consecutive în mod continuu (glissando), iar sistemul ar putea decide că trecerea de la "sol" la "la"

[1] Vedeţi şi articolul despre Computer Vision PC Report din iulie.

s-a făcut în alt moment decît intenţiona utilizatorul, note fictive pot apărea, de obicei armonicele notei cîntate, pentru că am văzut că sunetele emise nu sunt pure. Extragerea notelor este una din componentele în care s-au înregistrat progrese substanţiale în ultima vreme, şi pe care o putem considera rezolvată satisfăcător din punct de vedere ingineresc. Programe precum Digital Ear `http://digitalear.iwarp.com` şi Autoscore `http://www.wildcat.com/Site/Homepage.htm` au ajuns la performanţe uimitoare.

O a treia problemă este cea a bazei de date. Căutarea necesită o bază de date cu toate cîntecele ce se doresc recunoscute, stocate în format simbolic. La ora actuală nu există nici o astfel de colecţie care să fie suficient de comprehensivă pentru a fi cu adevărat utilă în aplicaţii practice.

4 Soluţii

Să ne reamintim că încercăm să definim o noţiune de *distanţă* între două fragmente muzicale, care măsoară similaritatea dintre ele.

Să încercăm să vedem ce soluţii există pentru toate aceste probleme. Prima problemă constă în inacurateţea reproducerii fragmentului de către utilizatorii sistemului. De aceea, sistemul trebuie sa folosească o măsură a similarităţii care să aibă următoarele proprietăţi:

- să fie invariantă la tempo (deci dacă melodia este cîntată mai lent sau mai rapid trebuie să fie totuşi recunoscută)
- să fie invariantă la gama muzicală (adică dacă melodia este cîntată cu două note mai sus să o recunoască totuşi)
- să fie tolerantă la note false

Date două fragmente de melodii, putem măsura similaritatea dintre ele prin *compararea graficelor lor*. Problema similarităţii a două fragmente de melodii se transformă astfel într-una de comparare a două funcţii treaptă.

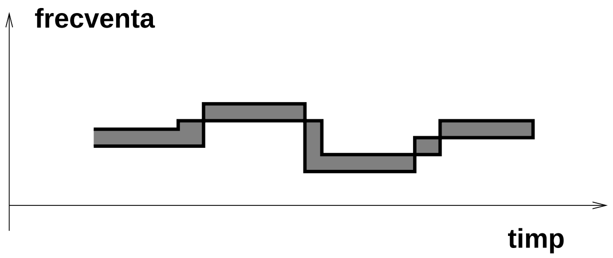

Figura 3: "Distanţa" dintre două melodii reprezentate ca funcţii treaptă se poate măsura prin aria dintre ele.

Să presupunem că cele două fragmente au aceeaşi lungime. Graficele corespunzătoare vor avea şi ele aceeaşi lungime. Dacă le vom vizualiza suprapuse, o bună măsură a distanţei dintre ele este aria dintre cele două grafice, ca în figura 3.

Cînd calculăm distanţa dintre grafice, trebuie să *translatăm* unul din grafice pe verticală, astfel încît sa minimizăm aria dintre ele. În felul acesta distanţa este invariantă la schimbarea gamei şi, în acelaşi timp, tolerantă la note false.

Pentru a obţine invarianţă la tempo, încercăm să dilatăm unul din fragmente cu diverşi factori, pentru a minimiza aria dintre cele doua funcţii.

Iată o schiţă a algoritmului pe care Cristi l-a implementat, care calculează distanţa dintre un fragment cîntat şi o melodie monofonică (care are o singură linie melodică, fără acompaniament):

```
pentru diverşi factori de scalare între 0.5 şi 2
    scalează fragmentul cu factorul curent
        translatează graficul lui de-a lungul graficului melodiei (pe orizontală)
        pentru fiecare poziţie intermediară
            translatează fragmentul pe verticală pentru a minimiza aria dintre cele două grafice
raportează cea mai mică arie găsită în acest proces
```

Funcţionarea acestui algoritm este ilustrată în figura 4.

4.1 Alte probleme de inginerie

Am rezolvat astfel problema teoretică cea mai spinoasă, de a defini o distanţă. Aceasta se pretează la o implementare relativ simplă, dar mai avem de rezolvat cîteva alte probleme neplăcute înainte de a face acest sistem cu adevărat practic.

Prima dintre ele este faptul că melodiile nu sunt, în general, monofonice (în vreme ce reproducerile oamenilor sunt). De aceea, înainte de a putea aplica algoritmul nostru, trebuie să transformăm fiecare cîntec într-o melodie monofonică. Acest lucru este făcut făcînd o analiză spectrală a muzicii şi eliminînd toate notele în afară de cea cu energia maximă[2].

A doua problemă constă în extragerea notelor din fişiere de tip `wav` (conversia de la o înregistrare la un format simbolic). În mod cert, calitatea programelor ce convertesc `wav` la MIDI a crescut foarte mult, pentru fişiere monofonice. Distanţa pe care o folosim însă este tolerantă şi erori introduse de această conversie: fragmentare sau alipire de note, trecerea de la o notă la alta la momente inoportune.

A treia problemă constă în lipsa de cîntece în format simbolic. Un mod de a rezolva această problemă este de culege fişiere MIDI de pe Web. Pe web există cel puţin 100 000 de astfel fişiere, suficient pentru a construi o bază de date respectabilă folosind un "păianjen" (web spider, crawler), asemănător celor folosite de motoarele de căutare. Se estimează că 99.99% dintre cererile de căutare de cîntece pe Internet se referă la mai puţin de 10000 de melodii diferite.

Deşi folosirea MIDI este atractivă datorită numărului mare de fişiere disponibile, există şi dezavantaje: marea majoritate a acestor fişiere sunt create de către amatori, ceea ce duce la o calitate îndoielnică a reproducerii originalului. Aşa cum am menţionat mai sus, prezenţa unor multiple canale şi a mai multor sunete simultane în fiecare canal este un impediment în folosirea acestor fişiere; cu toate acestea, din punct de vedere practic fişierele MIDI reprezintă un compromis excelent, fiind într-un format uşor de prelucrat, şi în număr foarte mare.

Acestea fiind zise să aruncăm o privire asupra sistemului în curs de dezvoltare în cadrul departamentului de calculatoare al universităţii Rutgers, descris în figura 5. Sistemul este format din trei componente majore. Prima colectează fişiere MIDI de pe web într-un proces continuu, le clasifică şi le adaugă la baza de date. A doua componentă preia cererile de căutare pe bază de

[2]Aceasta nu este neapărat cea mai bună metodă pentru a detecta "melodia principală", dar este foarte simplă şi funcţionează bine în practică.

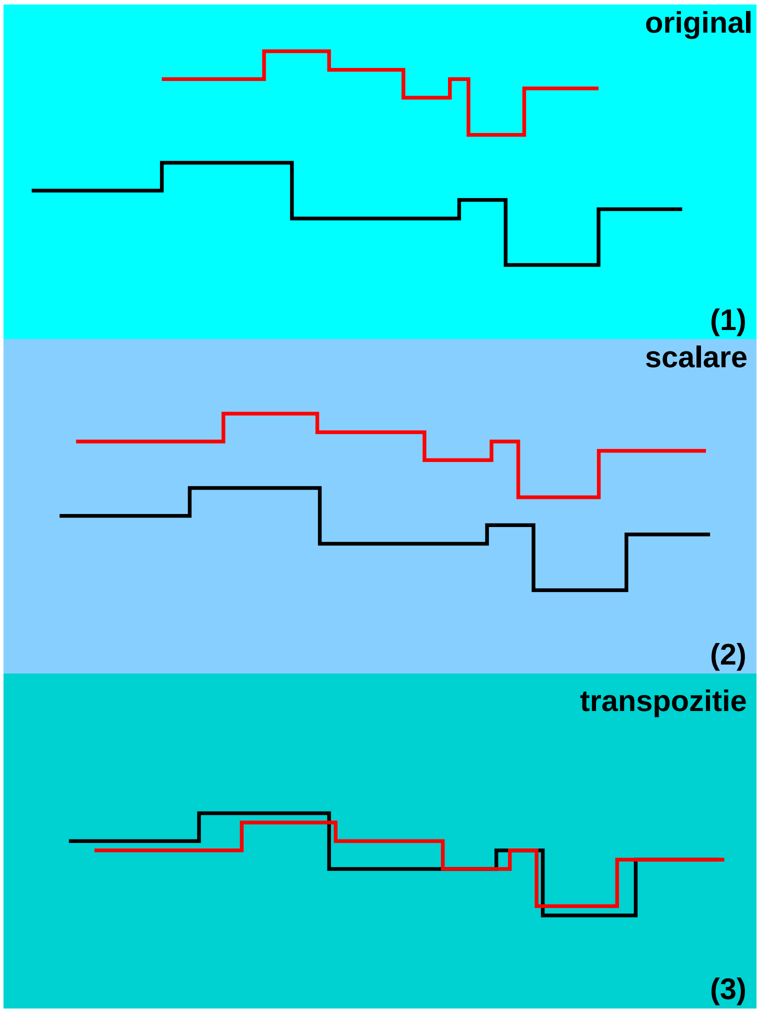

Figura 4: Funcţionarea algoritmului de calcul a distanţei: (1) date fiind o melodie (linia neagră) şi o întrebare (linia roşie) exprimate prin funcţii treaptă, trebuie să găsim "potriveala" maximă dintre ele. În prima fază (care nu e ilustrată) întrebarea este translatată în timp pentru a găsi locul unde apare în melodie (nu neapărat la început), apoi (2) întrebarea este scalată în timp, pentru a potrivi tempo-ul cu cel al melodiei, şi în fine (3) este scalată în frecvenţă pentru a fi la aceeaşi "înălţime" cu piesa. Toţi cei trei paşi sunt făcuţi pînă cînd distanţa (aria dintre cele două curbe) este minimizată; calculul distanţei este deci un *proces de căutare.*

text (autor, cîntec, album, versuri) şi execută o căutare textuală în baza de date, folosind tehnici standard de biblioteci digitale şi informaţiile simbolice prezente în fişierele MIDI. A treia componentă preia cererile de căutare pe bază de fragmente cîntate, le transformă în fişiere MIDI folosind o subcomponentă de extragere a notelor, apoi caută fragmentul respectiv în cîntecele din baza de date folosind algoritmul de mai sus.

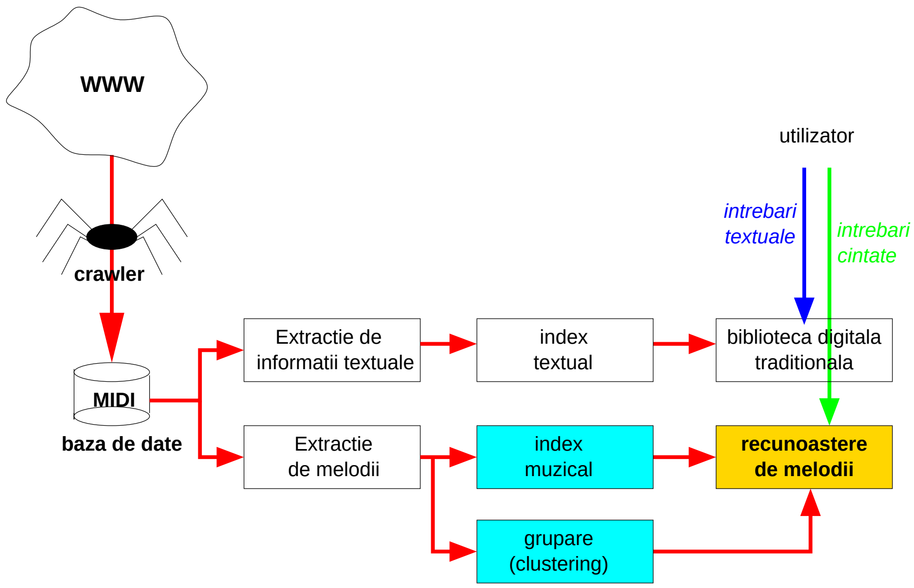

Figura 5: Sistemul `humwistle`, `http://www.humwistle.com` în întregime. Componentele indicate cu albastru sunt în curs de dezvoltare. O colecţie mare de fişiere MIDI este extrasă de pe web. Informaţiile textuale inserate în corpul fişierelor MIDI sunt extrase şi folosite pentru etichetarea şi indexarea fişierelor. Componenta aurie este cea descrisă în acest articol şi este partea cea mai inovatoare a sistemului, care permite căutarea după melodii cîntate.

5 Experimente

Rezultatele obţinute cu acest sistem sunt încurajatoare. Am efectuat experimente pe un set de 10 000 de cîntece, din totalul de 100 000 existente în baza de date. Mai mulţi subiecţi au cîntat 30 de fragmente muzicale diferite, fiecare între 4 şi 22 de secunde în lungime. Sistemul a returnat pentru fiecare fragment cele mai potrivite 20 de melodii.

Sistemul a identificat cel mai adesea cu succes diferite versiuni ale cîntecului real, din care făcea parte fragmentul (chiar dacă cîntecul era substanţial diferit, cu ritm sincopat sau cu variaţiuni). Cîntecul original, din care făcea parte fragmentul, a fost aproape mereu prezent între primele 20 de posibilităţi, cel mai adesea pe primul loc.

Un al doilea set de experimente a avut ca obiectiv evaluarea performanţei subiecţilor umani pentru acelaşi test: cît de buni sunt oamenii la recunoaşterea unui cîntec după un fragment cîntat de cineva. Am selectat şase fragmente dintre cele 30. Subiecţii umani au pornit recunoaşterea de la fragmentul transpus în formă simbolică MIDI, deci după pitch tracker; în felul acesta au primit aceleaşi erori introduse de pitch tracker pe care le-a primit şi sistemul automat. Subiecţii nu au avut constrîngeri de timp şi li s-a permis să asculte fragmentele de ori cîte ori au dorit. Am considerat că un subiect a recunoscut cîntecul fie dacă a putut spune numele, fie dacă a putut să fredoneze cîteva note în continuarea fragmentului. Nici unul din subiecţi nu a fost muzician profesionist. Toate fragmentele muzicale erau cunoscute subiecţilor.

În medie subiecţii au recunoscut 1,5 cîntece din 6 posibile. Numărul maxim de cîntece recunoscute a fost 4, iar numărul minim zero. În schimb, sistemul a reuşit să identifice 4 din 6. Concluzia acestui experiment este că recunoaşterea cîntecelor este o problemă grea chiar şi pentru oameni, şi este plauzibil ca în viitor calculatoarele să devină mai bune decît omul în acest domeniu.

Calculele necesare pentru identificarea unui cîntec sunt foarte costisitoare: implementarea curentă necesită aproximativ şapte ore şi jumătate pentru a răspunde la o întrebare, şi aceasta folosind baza de date redusă, de 10000 de melodii. De aceea eforturile de cercetare se concentrează în găsirea unui algoritm mai rapid. De asemenea, trecerea de la 10000 de cîntece la un milion de cîntece este deosebit de grea: indiferent de viteza algoritmului de calcul al distanţei, el tot va trebui sa compare fragmentul cu un milion de posibili candidaţi. De aceea eforturile noastre se îndreaptă spre găsirea unor metode de *indexare* care să permită eliminarea din start a majorităţii cîntecelor, ca fiind foarte diferite. Algoritmi folosind tehnologii de inteligenţă artificială vor fi folosiţi pentru a grupa (clustering) cîntecele în categorii înrudite, care pot fi eliminate dintr-o dată.

6 Alte eforturi de cercetare în biblioteci muzicale

Proiectul humwistle nu este singurul care studiază problema creării şi accesării unei biblioteci digitale. Pare însă să fie cel care a rezolvat cu cel mai mare succes problema căutării, datorită funcţiei distanţă folosite pentru măsurători. În această secţiune enumerăm pe scurt alte cîteva proiecte, care sunt din cunoştinţele autorilor cele mai avansate în domeniu.

Cercetarea universitară în domeniul similarităţii melodiilor a început prin anii '70, dar sisteme practice nu au apărut pînă în 1995. Cauzele acestei stagnări sînt multiple; Internet-ul a explodat de curînd, deci cererea de căutare după conţinut are mult mai mulţi "clienţi". Marile companii de discuri au început să folosească Internet-ul de numai cîţiva ani. De asemenea, fişierele MIDI au devenit cu adevărat populare de curînd.

Asif Ghias şi echipa lui de la universitatea Cornell din SUA în 1995 au construit primul sistem experimental de recunoaştere a unui cîntec. Metoda folosită de ei constă în extragerea notelor din fragmentul cîntat; apoi fragmentul este transcris ca o secvenţă de simboluri U, D, sau S. O notă este transformată în U (up) dacă este mai înaltă decît nota anterioară, S (same) dacă este aceeaşi notă sau D (down) dacă este mai joasă. Melodiile din baza de date sunt reprezentate în aceeaşi manieră. Fragmentul este căutat în baza de date folosind un algoritm binecunoscut de comparare a unor şiruri de caractere, permiţînd mai puţin de k diferenţe (algoritmul Baesa-Yates).

Sistemul a demonstrat viabilitatea ideii, dar lăsa de dorit din punct de vedere practic: baza de date conţinea numai 183 de cîntece, iar algoritmul necesita din ce în ce mai multe note în fragmentul cîntat pe măsură ce baza de date creşte,[3] făcîndu-l nepractic pentru 10000 de cîntece.

De departe cele mai bune rezultate în domeniu au fost obţinute de McNab şi echipa sa, la universitatea Waikato din Noua Zeelandă. Ei au construit un sistem care a testat mai multe funcţii distanţă diferită. Baza lor de date, asamblată din partituri de cîntece populare germane şi chinezeşti, conţinea 10000 de melodii. Metoda de comparare care a dat cele mai bune rezultate constă în exprimarea fiecărei note ca o pereche (interval, durată), unde intervalul este distanţa faţă de nota precedentă. Cîntecele din baza de date erau codificate în acelaşi fel, iar distanţa folosită era "distanţa de editare". Distanţa de editare dintre două şiruri de caractere, *edit-distance*, este numărul minim de operaţii de inserare, ştergere sau înlocuire de caracter care trebuie aplicate primului şir pentru a-l obţine pe al doilea.

[3] Asta pentru că, dacă transformăm cîntecele doar în U, D, S, ele tind să semene mult între ele; pentru a le diferenţia e nevoie de din ce în ce mai multe note.

Sistemul a demonstrat performanţe uimitoare: atunci cînd fragmentul cîntat era chiar începutul cîntecului, 12 note erau de ajuns pentru a reduce numărul de melodii candidat la mai puţin de 10. Din păcate publicaţiile lor nu spun ce se întîmplă atunci cînd fragmentul de recunoscut poate fi oriunde în cîntec. Cercetări ulterioare sugerează că, în acest caz, numărul de note necesare pentru dezambiguare creşte din nou foarte mult, făcînd sistemul nepractic.

Un alt pionier al căutării pe bază de conţinut este J. S. Downie, de la universitatea din Western Ontario, Canada. El a observat faptul că atunci cînd baza de date creşte este prea costisitor să comparăm fragmentul cîntat cu toate melodiile: numărul de calcule ar fi prea mare. De aceea el a încercat să aplice tehnici clasice de indexare folosite în bibliotecile digitale cu text şi de motoarele de căutare de pe World Wide Web. El a definit "cuvintele" ca fiind secvenţe de patru sau cinci note consecutive şi apoi a construit un index foarte asemănător cu indexul de la sfîrşitul unei cărţi tehnice: pentru fiecare cuvînt, indexul indică locul lui de apariţie, pagina în cazul cărţii, cîntecul în cazul nostru. Un fragment de recunoscut este segmentat în note, apoi scris ca o succesiune de "cuvinte". Fiecare din aceste cuvinte apare în mai multe cîntece, iar, dintre acestea, cîntecul care apare în listele indicate de toate cuvintele este cel căutat.

Metoda funcţionează foarte bine şi mai ales rapid, atunci cînd fragmentele căutate sunt reproduceri fidele ale originalului. Din păcate, performanţa sistemului se degradează foarte mult atunci cînd o notă sau două sunt greşite, ceea ce, din nou, face sistemul nepractic.

Toate aceste încercări de a construi o bibliotecă digitală muzicală au avut un punct comun: modul lor de reprezentare a melodiilor era o secvenţă de simboluri discrete, ca un şir de caractere. Un simbol este fie nota, fie diferenţa în semitonuri faţă de nota anterioară, fie o pereche de un de număr şi o durată. Această reprezentare este problematică pentru construcţia unei bune măsuri de similaritate, deoarece o notă fragmentată, sau mai multe note concatenate schimba dramatic reprezentarea unui fragment, deşi, în esenţă, el diferă puţin de original. De aceea, noi am optat pentru reprezentarea prezentată mai sus, în care o melodie nu mai este o înşiruire de simboluri, ci mai degrabă o variaţie continuă în timp a înălţimii notei. Considerăm că aceasta este cheia care a permis obţinerea unor rezultate mai bune decît oricare din cele raportate pînă în prezent.

7 Evoluţii viitoare

Sistemul humwistle este disponibil pe Internet. În momentul de faţă numai căutarea textuală este accesibilă on-line. Cînd performanţele căutării pe bază de conţinut se vor îmbunătăţi suficient pentru o utilizare interactivă, ea va fi de asemenea disponibilă pe web.

Aplicaţiile comerciale ale sistemului sunt enorme: companiile de discuri ar putea crea astfel de baze de date, în care fiecare piesă în format `wav` sau PCM (de pe CD) este însoţită de o descriere MIDI, eventual extrasă automat. Utilizatorul poate interoga baza de date, iar sistemul descris poate identifica melodia MIDI corespunzătoare. Apoi utilizatorul poate cumpăra după dorinţă înregistrările piesei, sau partitura, extrasă direct din MIDI.

Muzicienii ar putea folosi un astfel de sistem pentru a detecta plagiarismul în cantităţi enorme de cîntece, căutînd temele din propriile lor compoziţii.

O aplicaţie interesantă acestei tehnologii este în calculatoare mobile, unde spaţiul ecran este foarte restrîns. Posibilitatea de a selecta cîntece spre ascultare prin cîntarea unui fragment ar putea fi de nepreţuit. Un player de MP3-uri conectat la reţeaua de telefonie celulară ar putea sa aducă cîntece la cerere, pornind de la scurte fragmente cîntate.

Imaginaţi-vă conducînd un automobil peste zece ani, care va include un player de MP3 care are suficientă memorie pentru a stoca toată colecţia dumneavoastră de cîntece. Cum aţi putea face pentru a selecta o melodie sau un album spre ascultare, fără să trebuiască să luaţi mîna de pe volan pentru a scotoci în cutia cu discuri? O soluţie elegantă ar fi să cîntaţi un fragment din cîntecul dorit. În plus, această soluţie nu necesită luarea mîinilor de pe volan.

8 Alte surse de informaţie

Pe Internet există o cantitate enormă de informaţie despre formatele digitale audio, şi mai ales despre MP3, care e la mare vogă. Folosiţi motorul dumneavoastră de căutare favorit.

Sistemul dezvoltat este disponibil on-line la `http://www.humwhistle.com`.

Cristian Frâncu şi Craig Nevill-Manning “Distance metrics and indexing strategies for a digital library of popular music”, International Conference on Multimedia and Expo, July 30 — August 2, 2000, New York City, USA, `http://sequence.rutgers.edu/~francu/Papers/icme00.pdf`.

Asociaţia internaţională pentru muzică pe calculator are liste foarte impresionante de resurse şi informaţii plecînd de la `http://www.computermusic.org/`.

Fraunhofer Gesellschaft `http://www.iis.fhg.de/amm/techinf` este institutul German care a creat MPEG 2 layer 3, sau MP3.