

COMPRESIA SECVENȚELOR VIDEO – STANDARDE MPEG

1. Noțiuni teoretice

1.1 Noțiuni introductive

MPEG (**M**oving **P**ictures **E**xperts **G**roup) este un comitet internațional de standardizare, specializat în standardizarea compresiei datelor audio și video. Pornindu-se inițial de la metode de compresie video pentru rate de 1.2 MBps având ca mediu de stocare CD-ROM-ul, standardele MPEG sunt în prezent cele mai utilizate instrumente pentru compresia audio-video de înaltă calitate.

În această lucrare de laborator se intenționează prezentarea succintă a ceea ce reprezintă standardul **MPEG-1** respectiv **MPEG-2**. Totodată se vor prezenta cu titlu informativ celelalte tipuri de standarde dezvoltate de MPEG pentru aplicații multimedia: MPEG-4, MPEG-7, AAC (Advanced Audio Compression) și DSM-CC (Digital Storage Media Command and Control).

Cele mai importante aplicații care au condus la dezvoltarea standardelor de compresie MPEG sunt:

- **Capacitatea de stocare pe disk** – MPEG 1 a fost standardul care a încercat să rezolve problema compresiei audio-video, astfel încât să se poată folosi ca mediu de stocare CD-ROM-ul (rate de transfer 1.2 MBps). Astfel pentru a comprima o secvență video de aproximativ 210 MBps (necomprimată - eșantioane pe 8 biți) este necesară o rată de compresie de aproximativ 1:200 pentru a putea obține o rată de transfer de 1.2 MBps folosită de un CD-ROM clasic. De curând au apărut alte medii de stocare. Discul optic DVD (Digital Versatile Disc – redenumire a mai vechiului Digital Video Disc) permite stocarea unei cantități mai mari de informație, la rate de transfer mult mai mari, care fac ca aceste dispozitive să fie comparabile cu mediile clasice de stocare a secvențelor video (casetă VHS, TV – clasică)
- **Transmisia semnalului video digital** – spectrul electromagnetic disponibil pentru distribuția materialelor audio-video este limitat, deci este utilă implementarea și dezvoltarea unor tehnologii de compresie pentru aplicațiile de transmisie digitale. Deja multe din țările lumii folosesc transmisia video digitală prin intermediul sateliților de comunicație.
- **Transmisia comutată a datelor video digitale** – presupune stabilirea unei căi de comunicare între sursa video (care poate fi un codor în timp real sau un sistem de stocare) și decodor. Pentru transmisia comutată digitală se poate folosi o tehnologie numită ADSL (asymmetric digital subscriber line), care permite distribuirea semnalului digital folosind rețeaua existentă telefonică. Există și aici o limitare a ratei depinzând de tipul modulației folosite.
- **Televiziunea de înaltă definiție** – HDTV (High Definition Television) – este aplicația similară televiziunii clasice care însă permite transmisia semnalului video la o calitate superioară. Folosirea tehnicilor de compresie a semnalului video, face ca banda folosită de un canal HDTV să fie comparabilă cu cea folosită de televiziunea clasică.
- **Aplicații multimedia** – dezvoltarea aplicațiilor multimedia folosind în special rețelele de comunicații s-a putut realiza numai prin folosirea compresiei datelor video digitale. Astfel folosirea unui standard unic de compresie pentru secvențele video a putut face ca diverse aplicații multimedia să poată fi transmise pe diverse tipuri de rețele.

Unul dintre elementele principale în dezvoltarea unui standard este interoperabilitatea. Un standard bun este un standard care poate fi implementat în cât mai multe aplicații indiferent de platforma sistem utilizată. Strategia folosită de către MPEG pentru standardizarea compresiei audio-video cuprinde câteva **principii** extrem de utile:

1. **Un stream MPEG** – este un șir care respectă toate regulile sintactice și semantice date de MPEG;
2. **Un codor MPEG** – este acel codor care produce un stream MPEG;
3. **Un decodor MPEG** – este un decodor care decodează toate stream-urile MPEG;
4. **Un codor MPEG bun** – este un codor care generează un stream MPEG cu o rată de compresie cât mai ridicată;
5. **Un decodor MPEG bun** – este acel decodor care procesează în condiții bune datele audio-video la dispozitivele de redare, și care poate corecta o parte din eventualele erori apărute în stream-ul MPEG.

1.2 Componentele standardului MPEG-1 și MPEG-2

Standardele MPEG-1 și MPEG-2 sunt împărțite în câteva părți:

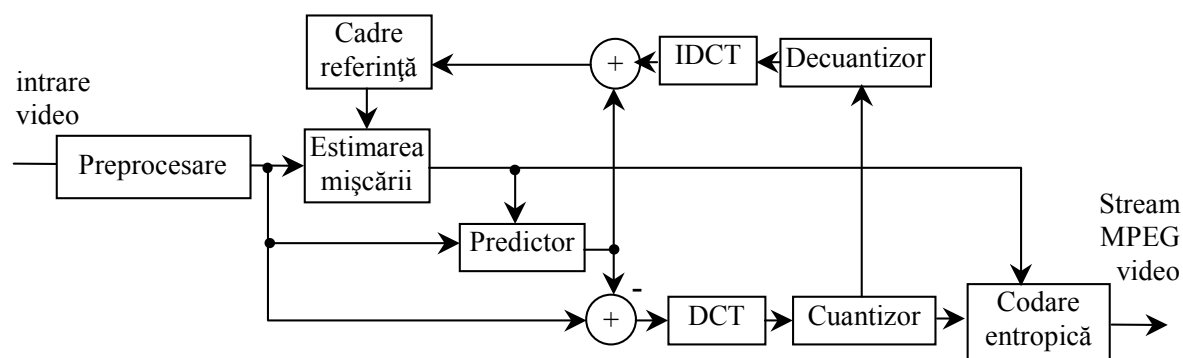
- Componenta “**system**” – descrie modalitățile în care diferitele tipuri de stream-uri (audio, video, sau date generice) sunt multiplexate și sincronizate.
- Componenta “**Video**” – definește compresia datelor video
- Componenta “**Audio**” – definește compresia datelor audio

Pentru a putea înțelege componenta “system” vom trata pentru început componentele “video” și “audio”.

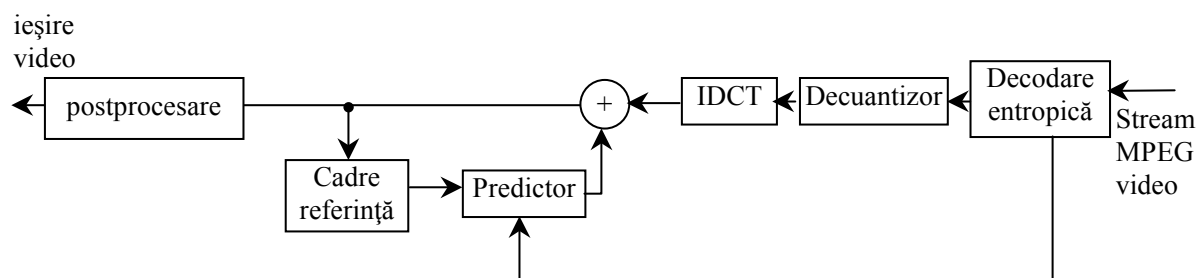
1.3 Componenta MPEG Video

1.3.1 Algoritmul de bază

În figura 1 este prezentată diagrama bloc a codorului MPEG (a), respectiv decodorului MPEG Video (b).



a) schema bloc a codorului MPEG Video



b) schema bloc a decodului MPEG Video

Figura 1 Schema bloc a codorului/decodului MPEG Video

Așa cum se poate observa din figura 1.a, intrarea video este prevăzută cu un bloc de *preprocesare* (interpolare și filtrare), urmată de blocul de *estimare a mișcării* folosit pentru determina *predictorul* imaginii (cadrlui) curente, din cadrele transmise anterior. Vectorii de mișcare reprezintă o parte de informație și sunt trimiși blocului de *codare entropică*. Predictorul este scăzut din fiecare bloc în parte, rezultând *eroarea de predicție* care este trimisă blocului *transformării cosinus discrete DCT*. Coeficienții DCT sunt cuantizați în blocul de *cuantizare* și apoi codați *entropic* pentru a putea fi transmiși. Totodată coeficienții cuantizați sunt folosiți pentru a reconstrui cadrele de referință. Aceștia sunt trimiși blocului de *cuantizare inversă* (decuantizare), urmat apoi de blocul *transformării cosinus inverse IDCT*, și adunate apoi cu eroarea de predicție. În felul acesta se obține un cadru de referință (asemănător cu cel obținut la decodare) folosit pentru viitoarea estimare a mișcării.

Decodorul MPEG video conține un bloc care realizează *decodarea entropică*, urmat apoi de blocul de caantizare inversă (*decuantizare*) și respectiv blocul transformării cosinus inverse *IDCT*. Informația obținută de la ieșirea acestui bloc este sumată cu cadrul de referință. La ieșirea sumatorului avem cadrul curent refăcut. Acesta este trecut în final printr-un bloc de *postprocesare* (interpolare și filtrare).

1.3.2 Reprezentarea secvențelor video

Secvențele video care vor fi prelucrate în procesul de compresie MPEG se compun dintr-o succesiune de *cadre* și *câmpuri* de *luminanță* și *crominanță*.

Reprezentarea pe cadre

Standardul MPEG-1 folosește doar reprezentarea pe cadre a semnalului video digital. Fiecare cadru constă din trei plane de pixeli, unul pentru *luminanță* (Y , alb-negru) și celelalte două pentru *crominanță* (câte unul pentru fiecare componentă C_R , C_B). Cele trei plane de pixeli se pot obține din semnalele primare R, G, B folosind același algoritm de descompunere ca și în televiziunea clasică.

Pentru MPEG-1 cele două plane de crominanță sunt eșantionate cu un factor egal cu 2 atât pe orizontală cât și pe verticală, relativ la planul de luminanță. Deoarece standardul MPEG-1 nu specifică modul de subeșantionare, se poate considera ca metodă implicită cea prezentată în figura 2.a.

Standardul MPEG-2 specifică subeșantionarea orizontală prescrisă de CCIR-601, ceea ce din punct de vedere spațial implică o reprezentare asemănătoare celei din figura 2.b.

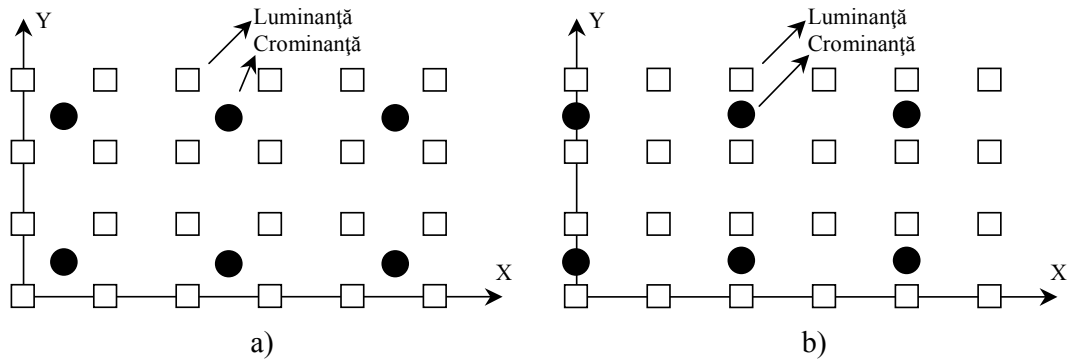


Figura 2 Relația spațială între luminanță și crominanță subeșantionată pentru:
a) standardul MPEG-1
b) standardul MPEG-2

Reprezentarea pe câmpuri

Standardul MPEG-2 este optimizat pentru o serie largă de aplicații, incluzând cele care folosesc reprezentarea pe câmpuri.

Câmpurile sunt create prin divizarea fiecărui cadru în două câmpuri (semicadre) întrețesute, ce conțin liniile (rândurile) pare respectiv impare dintr-un cadru. Câmpurile sunt întrețesute unul după altul și se succed la intervale de timp egale cu durata unui cadru. Se exploatează aici faptul că HVS (Human Visual System) este mai puțin sensibil la frecvențele spațiale și temporale înalte. Această metodă de întrețesere este în de fapt o metodă de compresie a semnalului video digital.

1.3.3 Predicția

Predicția temporală exploatează redundanța care există între două cadre succesive dintr-o secvență video. De exemplu, putem avea o secvență video în care fundalul nu se modifică, modificându-se doar poziția anumitelor obiecte din prim-plan. În acest caz prin aplicarea unui simplu algoritm de codare DPCM (Differential Pulse Code Modulation) se poate obține un grad mare de compresie. Acest algoritm de predicție folosește ca predictor pentru imaginea curentă, imaginea anterioară și se face o scădere între cele două imagini.

Totuși există și secvențe video în care întregul conținut al imaginii se modifică de la un cadru la altul. În astfel de cazuri se poate aplica un algoritm de estimare al mișcării obiectelor, care să poată fi folosit pentru predicția imaginii curente.

Nu întotdeauna algoritmul de estimare al mișcării folosit pentru predicția cadrului curent este eficient. Spre exemplu, va fi greu să determinăm particularitățile de mișcare în cazul în care o imagine se rotește sau este zoom-ată.

Compensarea mișcării

Modelul de mișcare folosit în standardul MPEG este cel al translației blocurilor de pixeli. Pentru simplitate s-a ales dimensiunea blocului ca fiind 16x16 pixeli, pentru componenta de luminanță și 8x8 pixeli (datorită subeșantionării cu 2) pentru componenta de crominanță.

"Reuniunea" dintre un bloc de 16x16 pixeli din planul de luminanță și blocurile de 8x8 pixeli corespunzătoare din planurile de crominanță formează un **macrobloc**.

Compensarea mișcării constă din luarea unui macrobloc din cadrul (imaginea) curent și determinarea offset-ului spațial al acestuia față de poziția din cadrul de referință. În felul acesta macroblocul curent poate fi prezis din cadrul de referință. Offset-ul se numește **vector de mișcare**. Algoritmii de compensare a mișcării folosesc vectorul de mișcare pentru a determina blocul de predicție din cadrul de referință. Odată găsit acest bloc se face diferența dintre cele

două blocuri și eroarea de predicție (după aplicarea DCT și cuantizării) este trimisă împreună cu vectorul de mișcare blocului de codare.

Vectorul de mișcare obținut se aplică în mod direct pentru a determina predictorul blocului de luminanță. Pentru a determina predictorii blocurilor de crominanță vectorul este scalat la jumătate (datorită subeșantionării inițiale) pe ambele direcții (verticală și orizontală).

Procesul de determinare, a celui mai bun vector care să fie folosit pentru prezicerea blocului curent, este cel mai complex din lanțul de prelucrare MPEG. În acest punct însă se obține cea mai mare compresie a secvenței video. Există mai multe metode de alegere a vectorului de mișcare dintr-o arie de căutare prestabilită. În practică cel mai folosit criteriu este cel al determinării erorii medii absolute (MAE – Mean absolute error) dintre macroblocul curent și cel determinat de vectorul de mișcare din cadrul precedent. Ca și algoritmi de estimare a mișcării, cel mai folosit în practică este *algoritmul de căutare piramidal* pentru o arie de căutare de 32x32 pixeli. Utilizarea unei astfel de dimensiuni pentru aria de căutare reprezintă un compromis între *precizia estimării* (care este cu atât mai exactă cu cât aria de căutare este mai mică) și *rata de bit* necesară transmiterii vectorilor de mișcare. În afară de algoritmul de estimare al mișcării piramidal se mai pot folosi și alți algoritmi de mișcare cum ar fi: *algoritmul de căutare bidimensional logaritmice*, *algoritmul de căutare exhaustivă*, *algoritmul de căutare pe direcții conjugate*, *algoritmul de căutare paralel*.

Tipuri de cadre

Una din principalele probleme care necesită a fi rezolvate o reprezintă accesul aleator la cadrele unei secvențe video. Accesul aleator la o imagine este destul de dificil de realizat în cazul unei secvențe de cadre care folosește compensarea mișcării. Ca și în cazul modulației delta sau a codării DPCM, secvența recepționată constă dintr-un șir de informații despre modificările care apar de la o imagine la alta. Deci pentru a putea accesa o imagine trebuie decodată întreaga secvență de imagini. Totodată în cazul apariției unei erori în transmisie imaginea nu mai poate fi refăcută cu exactitate, iar eroarea se propagă în tot lanțul de cadre.

Pentru a înlătura aceste neajunsuri există două posibilități:

- **procesul de refresh** – care presupune compresia unei imagini din secvența video care să nu folosească predicția. În acest fel se permite decodurului să se resincronizeze în acel punct.
- **predicția cu întreruperi** – are ca scop eliminarea erorilor de predicție și a erorilor care pot apărea în lanțul de transmisie și se pot propaga.

Folosind cele două strategii se rezolvă o altă problemă și anume aceea a accesului aleator la cadrele unei secvențe video. De exemplu, dacă vrem să vizualizăm cadrul 53 dintr-o secvență și știm că fiecare al zece-lea cadru, pornind de la zero, este codat fără predicție, atunci se decodează direct cadrul 50, urmat de decodarea cadrelor 51, 51 și respectiv 53. În felul acesta pentru a accesa un cadru, nu va trebui să decodăm mai mult de 10 cadre.

Pentru a facilita implementarea celor două concepte standardul MPEG a definit trei tipuri de cadre după cum urmează:

- cadre de tip **I**, cadre ce se codează fără a folosi tehnica de predicție. Practic aceste cadre se codează asemănător cu o imagine statică codată JPEG.
- cadre de tip **P**, care sunt imagini prezise și folosesc predicția cu salt înainte față de un cadru de referință ce poate fi un cadru I, sau P.
- cadre de tip **B**, cadre interpolate, folosind două cadre de referință de tip I sau P.

În figura 3 este prezentată o succesiune de cadre dintr-o secvență video și tipurile de predicție folosite pentru fiecare cadru în parte. Cadrele de tip I, și P se mai numesc cadre sau imagini ancoră deoarece ele vor fi folosite ca și cadre de referință pentru predicția celorlalte tipuri de cadre.

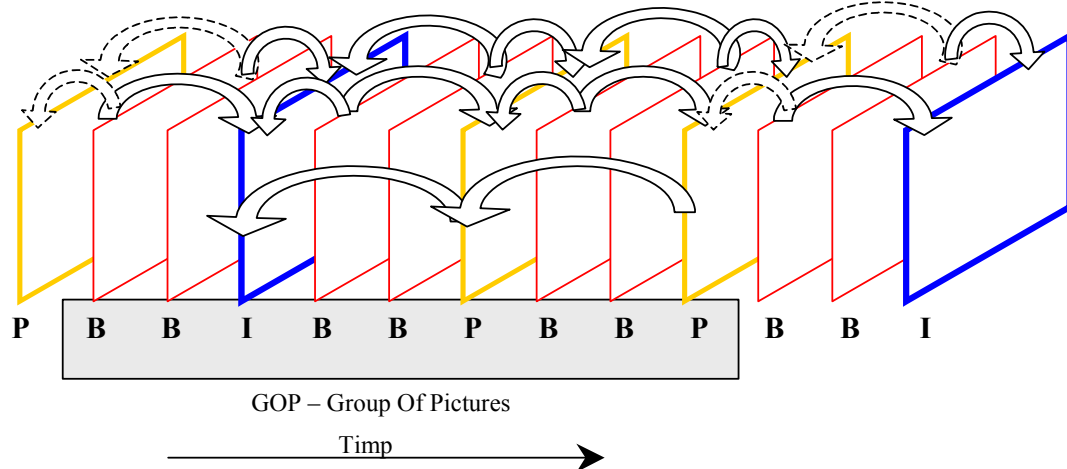


Figura 3 Tipurile de cadre și predicția folosită

Afișarea și ordinea de transmisie

Așa după cum se observă din figura 3, pentru a putea reface cadrele de tip P, și B avem nevoie de cadrele de referință corespunzătoare. Pentru a nu stoca informația în mod neeficient la decodor, ordinea de transmisie a cadrelor MPEG, va fi diferită, astfel încât orice cadru recepționat să poată fi imediat decodat și afișat. În felul acesta se reduce capacitatea de memorare necesară la decodare.

În figura 4 este exemplificat ordinea de transmisie a cadrelor MPEG și ordinea de decodare a acestora.

Putem defini aici termenul de GOP – Group of Pictures – ca fiind o secvență continuă de cadre, care începe cu un cadru de tip I (inclusiv) și se termină cu următorul cadru de tip I (exclusiv). Această definiție este aplicabilă pentru secvența cadrelor așa cum sunt ele pregătite pentru transmisie. Evident la afișare (figura 3) ordinea este cu totul alta. În exemplul prezentat în figura 4 un GOP este definit de cadrele: I_0 , B_{-2} , B_{-1} , P_3 , B_1 , B_2 , P_6 , B_4 , B_5 . În exemplul dat în figura 3 GOP – conține o configurație de 2 cadre B între două cadre ancoră (I,P) și două cadre P. Această configurație poate fi însă realizată și cu mai multe cadre B și/sau P.

Timp	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
Ordinea la codare	P_{-3}	B_{-2}	B_{-1}	I_0	B_1	B_2	P_3	B_4	B_5	P_6	B_7	B_8	I_9
Ordinea la transmisie	I_0	B_{-2}	B_{-1}	P_3	B_1	B_2	P_6	B_4	B_5	I_9
Memorie decodare 1	P_{-3}	P_{-3}	P_{-3}	P_3	P_3	P_3	P_3	P_3	P_3	I_9
Memorie decodare 2	I_0	I_0	I_0	I_0	I_0	I_0	P_6	P_6	P_6	P_6
Ordinea de afișare	P_{-3}	B_{-2}	B_{-1}	I_0	B_1	B_2	P_3	B_4	B_5	P_6

Figura 4 Ordinea de codare, transmisie și decodare a cadrelor dintr-o secvență video
Secvențele încadrate $I_0 \dots B_5$ reprezintă un GOP

Așa cum sunt definite cadrele de tip B, pentru a le reface se folosește predicția bidirecțională. În general pentru refacerea primelor două cadre de tip B dintr-un GOP (în cazul nostru B_{-2} , respectiv B_{-1}) este nevoie atât de cadrul I care urmează cât și de cadrul P anterior aflat în GOP-ul precedent. Deci dacă am dori să începem decodarea de la începutul unui GOP, adică cu cadrul I_0 , atunci cadrele B_{-2} , B_{-1} , nu ar putea fi decodate corect lipsind cadrul P_{-3} . Standardul

MPEG rezolvă acest neajuns prin introducerea conceptului de GOP – închis, care stipulează că toate cadrele B care preced un cadru de tip I dintr-un GOP nu vor putea folosi ca și ancoră un cadru dintr-un GOP anterior (în cazul nostru cadrul P₃). În figura 3 am arătat acest concept prin punctarea săgeților care fac legătura dintre un GOP la altul. De fapt nu se face predicție între cadre. În felul acesta se aplică conceptul de predicție cu întreruperi.

Pornind de la definirea tipurilor de cadre, se poate stabili numărul biților folosiți pentru reprezentarea fiecărui tip de cadru. Având în vedere faptul că imaginile de B nu vor mai fi folosite pentru predicția altor imagini, acesta vor putea fi reprezentate folosind un număr minim de biți/pixel. Nu același lucru se întâmplă pentru cadrele I și P care sunt folosite ca imagini de referință, și deci necesită o calitate superioară pentru menținerea detaliilor imaginii. S-a stabilit că un raport de 5:3:1 pentru reprezentarea cadrelor I, P, respectiv B este satisfăcătoare, mărindu-se în acest fel raportul de compresie. Evident acest raport poate fi modificat în funcție de dinamica secvenței video.

1.3.4 Transformarea DCT

După procesul de compensare a mișcării, informația obținută (eroarea de predicție) este transformată folosind transformata cosinus discretă DCT – Discrete Cosine Transform. Pentru a aplica DCT, macroblocul este împărțit în 6 blocuri de 8x8 pixeli, 4 corespunzătoare planului de luminanță, și câte unul pentru fiecare componentă de cromaticitate. În figura 5.a este prezentată descompunerea blocului de luminanță pentru standardul MPEG-1 și MPEG-2 varianta de reprezentare pe cadre. În figura 5.b este prezentat modul de descompunere a blocului de luminanță pentru standardul MPEG-2 varianta de reprezentare pe câmpuri.

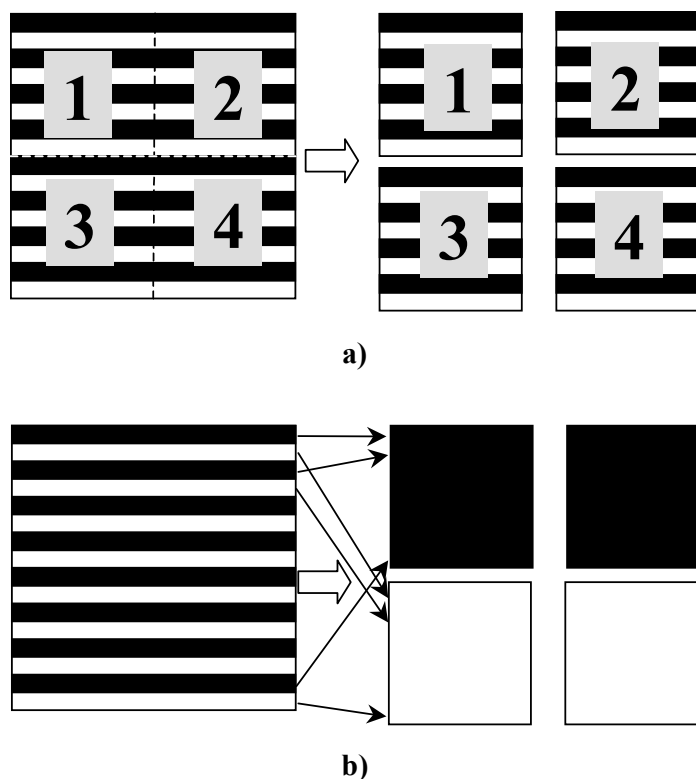


Figura 5 Descompunerea blocului de luminanță (16x16) în blocuri de 8x8 pixeli
a) pentru reprezentarea pe cadre
b) pentru reprezentarea pe câmpuri

1.3.5 Cuantizarea

După blocul transformării DCT, coeficienții obținuți sunt cuantizați folosind matricile de cuantizare din tabelele 1 (pentru cadrele codate fără predicție) respectiv tabelul 2 (pentru cadrele codate cu predicție). În urma procesului de cuantizare, acei coeficienți care sunt irelevanți din punct de vedere al calității imaginii vor fi practic eliminați. Se poate observa în tabelul 1 că primul element (stânga-sus) nu are valoare de cuantizare. Acest element corespunde cu coeficientul DC din matricea coeficienților DCT, și deci fiind cel mai important coeficient nu se cuantizează.

*	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Tabelul 1 Matricea de cuantizare pentru cadre ce nu folosesc predicția

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

Tabelul 2 Matricea de cuantizare pentru cadre care folosesc predicția

1.3.6 Codarea entropică

Atât coeficienții cuantizați cât și celelalte tipuri de informații (tipul de predicție, vectorii de mișcare, etc.) vor fi codați entropic pentru a se obține o rată de bit cât mai mică. Procesul codării entropice conține trei etape:

- conversia 2D la domeniul unidimensional
- codarea lungimii curselor (codarea RLC)
- codarea Huffman

Conversia 2D -> 1D

După cuantizare o mare parte din coeficienții DCT vor avea valoarea zero, așa cum se poate observa și din figura 6. Pentru a exploata acest aspect (frecvențele înalte sunt zero) în procesul de codare RLC, trebuie realizată o conversie a blocului 8x8 pixeli într-un șir unidimensional de coeficienți în care coeficienții nuli să fie grupați.

Pentru a realiza acest deziderat se pot folosi două tehnici de conversie:

- scanarea zig-zag (utilizată atât pentru MPEG-1 cât și pentru MPEG-2)
- scanarea verticală (utilizată în MPEG-2 – este mai eficient pentru cazul în care se folosește reprezentarea pe câmpuri)

În figura 7 a, b sunt prezentate traicitoriile folosite pentru conversia blocului 8x8 de coeficienți cuantizați.

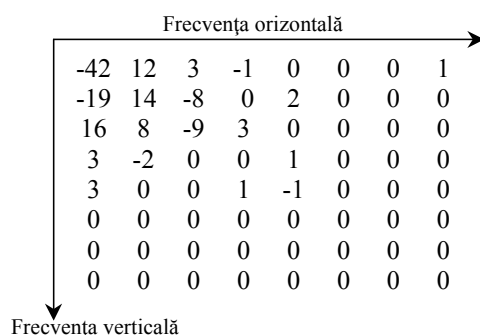
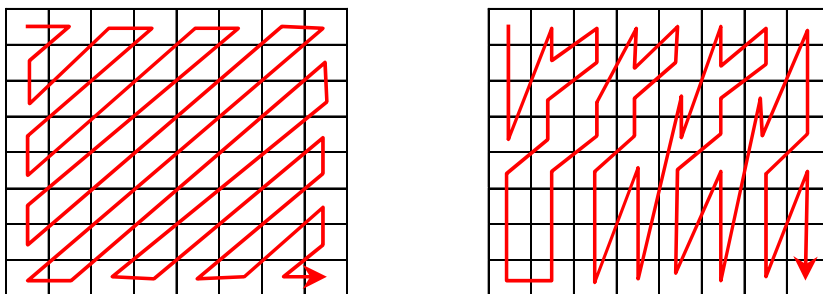


Figura 6 Bloc 8x8 coeficienți DCT cuantizați



a) b)
Figura 7 Tipuri de scanări pentru conversia 2D -> 1D:

- a) scanare zig-zag
 b) scanare verticală

Dacă aplicăm scanarea zig-zag pentru coeficienții DCT din figura 6 obținem secvența:

-42 12 -19 16 14 3 -1 -8 8 3 3 -2 -9 0 0 0 2 3 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 -1 0 ... 0

Codare lungimii curselor

După conversia 2D->1D se obține un vector unidimensional de dimensiune 64. Acest vector este trimis blocului de codare a lungimii curselor de zero. Cursa se definește ca fiind zero sau mai mulți de zero urmați o singur coeficient nenul. Fiecare astfel de pereche obținută va reprezenta un simbol pentru codul Huffman modificat. Dacă codăm secvența care se obține după scanarea zig-zag a coeficienților din figura 6 obținem codul RLC:

(0, -42), (0, 12), (0, -19), (0, 16), (0, 14), (0, 3), (0,-1), (0,-8), (0, 8), (0,3), (0,3), (0,-2), (0,-9), (3,2), (0,3), (10, 1), (2,1), (0,1), (6, -1),(EOB).

De notat faptul că simbolul (EOB) – cod special, semnifică faptul că nu mai există coeficienți nenuli. In felul acesta se elimină informația redundantă din blocul coeficienților DCT.

Codarea Huffman

Așa după cum se cunoaște codarea Huffman realizează o codare entropică optimală pentru un alfabet dat. Totuși în cazul în care alfabetul este prea mare este prea costisitoare implementarea unui astfel de algoritm. Pentru a folosi însă avantajele oferite de acest mod de codare, s-a dezvoltat un algoritm de codare Huffman modificată. Acest algoritm folosește codarea Huffman clasică pentru un număr (relativ mic) de simboluri (**simboluri de categoria I**) care au ce mai mare probabilitate de apariție. Celelalte simboluri (**simboluri de categoria II-a**) sunt codate cu prefix. Adică dacă apare un simbol ce nu este cuprins în prima grupă, acesta se codează trimițând prefixul, urmat de simbolul explicit. Introducerea codului de prefix va crește rata de bit a codului, dar dacă probabilitatea de apariție a acestor simboluri este suficient de mică atunci această creștere este nesemnificativă.

1.4 Componenta MPEG Audio

Standardul de compresie audio MPEG definește o serie de algoritmi (pentru diferitele tipuri de materiale audio – vorbire, muzică, efecte speciale) bazați pe compresia pe subbenzi folosind tehnici ce exploatează proprietățile sistemului auditiv uman.

Pentru standardul audio se definesc o serie de *nivele*, care pot fi folosite de diversele aplicații astfel încât să fie eficiente din punct de vedere al costului. Astfel putem defini 3 nivele, ordonate de la 1 la 3 în ordinea complexității. Astfel nivelul 1 folosește o rată de 192 kbps (dar folosește o codare simplă), nivelul 2 este utilizat pentru rate de 128 kbps, și în sfârșit nivelul 3 folosește rate de 64 kbps. În aplicațiile curente se folosesc unul din primele două nivele. Chiar dacă rata nu este atât de mică ea este neglijabilă în comparație cu rata obținută de componenta video.

1.4.1 Algoritm de bază

În figura 8 este prezentată schema bloc a algoritmului de codare/decodare MPEG Audio. Descompunerea în subbenzi separă semnalul audio de intrare în benzi multiple de frecvență. Fiecare subbandă este scalată și cuantizată (factorul de scalare diferă de la o subbandă la alta). În paralel se face o analiză în domeniul frecvență pentru a determina pasul de cuantizare specific fiecărei subbenzi în parte. Eșantioanele sunt apoi codate împreună cu informații suplimentare folosind codarea Huffman și apoi transmise. La decodare informația este demultiplexată și decodată. Eșantioanele sunt apoi decuantizate. La ieșirea cuantizatorului se aplică un factor pentru a rescala amplitudinea eșantioanelor corespunzătoare fiecărei subbenzi la valoarea inițială. În final semnalul este compus prin compunerea tuturor subbenzilor de frecvență.

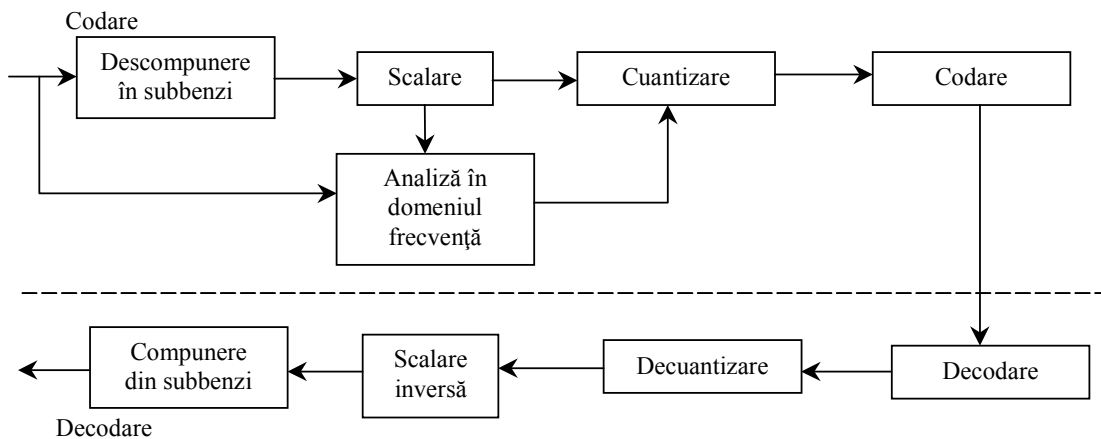


Figura 8 Schema bloc pentru algoritmul *MPEG Audio*

1.5 Componenta MPEG – System

Are rolul de multiplexare și sincronizare a tuturor informațiilor dintr-un stream MPEG astfel încât să se poată decoda corect un material multimedia.

Nivelul System implementat pentru MPEG-1 are în vedere faptul că acesta este folosit pentru dispozitivele de stocare și pentru aplicațiile care rulează pe calculatoare PC, rezultând pachete lungi de date de lungime variabilă.

În cazul standardului MPEG-2 avem două versiuni ale nivelului system:

- **stream-ul program**, care are aceeași concepție de construcție ca și cel utilizat în MPEG-1
- **stream-ul transport**, în care se folosesc pachete scurte de date de lungime fixă, facilitându-se în felul acesta operarea pe canale de transmisie cu zgomot și de asemenea pentru a facilita transmisia pentru aplicații de tip broadcasting.

2. Desfășurarea lucrării

2.1. Aparatură utilizată

Calculator PC - Pentium

2.2 Compresia unei secvențe AVI.

Se deschide fișierul *tv1.par* din directorul `c:\student\` folosind editorul **notepad.exe**. Acest fișier conține toți parametrii necesari programului **code.exe** care va realiza conversia din format AVI (de tip MJPEG) la fișier MPEG-1.

Cei mai importanți parametrii introduși în fișierul **tv1.par** sunt:

- INPUT_SEQ - specifică fișierul de intrare care trebuie să fie un fișier AVI în format M-JPEG
- MPEG_FILE - specifică fișierul MPEG de ieșire
- OUTPUT_SEQ - specifică fișierul de ieșire în care se salvează toate cadrele decodate
- DFD_SEQ - specifică fișierul cu diferențele dintre cadre
- FWDVF_SEQ - specifică fișierul în care se salvează vectorii de mișcare pentru estimarea de tip forward
- BWDVF_SEQ - specifică fișierul în care se salvează vectorii de mișcare pentru estimarea de tip backward
- RES_FILE - specifică fișierul de ieșire în care se vor salva datele despre conversia AVI-MPEG1
- NUM_P - specifică numărul de cadre de tip P folosite într-un GOP
- NUM_B - specifică numărul de cadre de tip B folosite într-un GOP
- NUM_FRAMES - numărul de cadre ce se codează și care vor intra în secvență MPEG-1
- BIT_RATE - rata de transfer

Se identifică și se notează valorile fiecărui parametru descris mai sus. Se deschide apoi o fereastră DOS și se dă comanda:

```
C:\student\code.exe tv1.par
```

În urma lansării acestei comenzi programul `code.exe` va genera 6 fișiere a căror semnificație am amintit-o deja. Cu ajutorul programului **vmpeg.exe** din directorul `c:\student\vmpeg\` se vizualizează fișierul MPEG generat. ATENTIE, fișierul MPEG generat nu va avea același număr de cadre ca și secvența AVI inițială dacă am setat o valoare specifică parametrului NUM_FRAMES din fișierul `tv1.par`!

Se modifică parametrii fișierului NUM_P și NUM_B, și se reface algoritmul de conversie. Ce se observă dacă se mărește numărul cadrelor B respectiv P?

În continuare vom utiliza fișierele definite ca variabile a parametrilor FWDVF_SEQ, respective BWDVF_SEQ, pentru a vizualiza vectorii de mișcare. Pentru aceasta vom folosi programul **hos2bmp.exe** care va converti fiecare plan de vectori de mișcare într-o imagine BMP. Astfel folosind comanda:

```
C:\student\hos2bmp.exe tv1fw.hos 20 30 vecfw
```

Vom obține o serie de 10 imagini vecfw*.bmp care reprezintă vectorii de mișcare pentru cadrele de la 20 la 30 folosind predicția înainte. Fișierul tv1fw.hos reprezintă valoarea parametrului FWDVF_SEQ din fișierul tv1.par.

În mod analog se procedează și pentru fișierul tv1bw.hos. Ce se poate spune despre secvența de imagini analizând imaginile ce conțin vectorii de mișcare?

Repetati algoritmul de conversie pentru secvența **vtest.avi**. Pentru aceasta scrieți propriul fișier de configurare de tip **vtest.par** care va conține toți parametrii descriși inițial.

Realizați o conversie în care NUM_FRAMES nu este setat (deci se codează toate cadrele secvenței) determinați raportul de compresie obținut prin aplicarea standardului MPEG-1. Studiați fișierul *.res și identificați parametrii trecuți în acest fișier.

3. Întrebări și probleme

1. Poate fi rulată o secvență codată MPEG-1 de pe un CD-ROM 2X (viteza standard 1X = 150kbps)?
2. Ce se întâmplă cu o secvență video la decodare dacă este eronat un GOP?
3. Explicați de ce în cazul unei transmisii de televiziune (folosind standardul MPEG-2) de proastă calitate imaginea recepționată prezintă efectul de blocking?
4. Cum se poate elimina efectul de blocking în cazul unei transmisii pe un canal cu zgomot?
5. Să se determine secvența de transmisie pentru o structură GOP formată din 2 cadre P, și 3 cadre B între două cadre ancoră.
6. De ce standardul MPEG folosește conceptul GOP-închis?
7. De ce un GOP conține numai un cadru de tip I? Am putea defini un GOP cu 2 sau mai multe cadre de tip I?
8. Se poate transmite un stream de cadre de imagine I,P,B în aceeași ordine ca la afișare? Care sunt avantajele și dezavantajele față de metoda adoptată de standardul MPEG?